Compute (Bridgend) Ltd

Reference and User Guide

SELCOPY/i Release 3.10

8 Merthyr Mawr Road, Bridgend, Wales UK CF31 3NH

Tel: +44 (1656) 65 2222
Fax: +44 (1656) 65 2227

CBL Web Site - http://www.cbl.com

This document may be downloaded from http://www.cbl.com/cblidoc.html

Contents

Contents

Contents

Contents

Compare Files/Libraries Menu (=7)

Contents

Contents

Contents

DB2 Utilities

Contents

DB2 Utilities

Contents

Documentation Notes

First Edition, May 2012

Information in this document details general features and functionality of the **SELCOPY Product Suite 3.10** component, **SELCOPY/i**.

This document replaces the previous edition of *SELCOPY/i Reference and User Guide* applicable to **SELCOPY Product Suite 3.00**, which is now obsolete.

SELCOPY Product Suite for z/OS, z/VM (CMS) and z/VSE operating systems, which includes SELCOPY, SELCOPY/i and CBLVCAT, is available for download and install from **http://www.cbl.com/selcdl.html**.

The following publications for SELCOPY Product Suite and its component products are available in Adobe Acrobat PDF format at CBL web page **http://www.cbl.com/selcdoc.html**:

- SELCOPY Product Suite Customisation Guide
- SELCOPY User Manual
- CBLVCAT User Manual
- SELCOPY/i Reference and User Guide
- SELCOPY/i Text Editor (CBLe) Manual
- SELCOPY/i Structured Data Editor Manual

At the time of publication, this document is believed to be correct. Where the program product differs from that stated herein, Compute (Bridgend) Ltd reserve the right to revise either the program or its documentation at their discretion. CBL do not warrant that upward compatibility will be maintained for any use made of this program product to perform any operation in a manner not documented within the user manual.

The following generic terms are used throughout this document to indicate all available versions and releases of IBM mainframe operating systems:

MVS	-	z/OS, OS/390, MVS/ESA, MVS/XA, MVS/SP, OS.
VSE	-	z/VSE, VSE/ESA, VSE/SP, DOS.
CMS	-	z/VM, VM/ESA, VM/XA, VM/SP.
All	-	All MVS, VSE and CMS operating systems.

Summary of Changes

This section is a summary of significant new features provided in SELCOPY/i Release 3.10.

First Edition (May 2012)

Primary Options Menu Support

Introduce primary options menu panels and sub-panels to perform most tasks supported by SELCOPY/i.

This manual has been completely restructured to so that tasks are documented in an order that reflects the panel hierarchy.

Enhancements to Compare Files Utility

The SELCOPY/i compare 2 files utility has been greatly enhanced to support the following:

◊ Formatted record compare
◊ Read-Ahead Synchronisation
◊ Sorted and Unsorted Key Synchronisation
◊ Hierarchical Compare

For details, see:

◊ *"Compare Files (=7.1)"*
◊ *"Record Synchronisation"*
◊ *"Compare Files Panels"*
◊ *"Basic Unformatted Compare Panel"*
◊ *"Extended Unformatted Compare Panels"*
◊ *"Formatted Compare Panels"*
◊ *"Compare Files Output"*
◊ COMPFILE

About SELCOPY/i

System Environment

SELCOPY/i is a full screen 3270 application which executes in any of the following environments:

◊ z/OS TSO/E.
◊ z/OS TSO/E as an ISPF application.
◊ z/OS as a stand-alone, multi-user VTAM application. (See Disaster Recovery below)
◊ z/VSE as a stand-alone, multi-user VTAM application.
◊ z/VM CMS.

General Functionality

The SELCOPY/i environment includes a compendium of tools and facilities that operate within a windowed environment within the 3270 display. The following is a selection of the supported features:

◊ Function rich text editor (CBLe) with both ISPF Edit and XEDIT compatibility.
◊ Structured data editor supporting COBOL and PL1 Copybooks.
◊ DB2 table editor.
◊ DB2 SQL execution.
◊ List facility includes DB2 objects, Datasets, DASD Volumes, VTOC files/extents, HFS files and ENQs.
◊ File Search and Update including support for copybook map.
◊ File Copy supporting mixed data set organisations and copybook remap.
◊ File and PDS/PDSE library Compare.
◊ SELCOPY interactive development environment and debugger.
◊ CBLVCAT interactive reports and VSAM data set tuning.

SELCOPY/i Environment

SELCOPY/i is a full screen 3270 interface which provides the user with a working environment whereby all tools and facilities (dialog panels, edit views, data set lists, etc.) are presented in windows within the 3270 display. Like PC workstation and UNIX based operating systems that support a windowed environment, SELCOPY/i includes the following features:

◊ Multiple overlapping window views.
◊ Window title and menu bars.
◊ Minimise, Maximise and Resize of displayed windows.
◊ Repositioning of displayed windows.
◊ Point-and-shoot buttons.
◊ Drop down and popup menus.

Disaster Recovery

SELCOPY/i has the same functionality when executing as a VTAM application as it does when running in user's TSO/E address space. Therefore, customising SELCOPY/i to execute as a stand-alone VTAM application from a recovery volume, would allow functions such as data editing, job submission, data set allocation and system navigation, even if ISPF and TSO/E are unavailable.

In addition to functionality included in the SELCOPY and CBLVCAT batch utilities, the systems programmer is provided with a powerful set of tools to assist in the data recovery process.

Getting started with SELCOPY/i

This chapter introduces end users to some basic SELCOPY/i concepts.

Starting the SELCOPY/i program

How SELCOPY/i is started depends on the environment in which it is to be executed.

OS	Environment	Command
MVS	TSO/E	Enter **SELCOPYI** at the READY prompt.
	ISPF	Enter **TSO %SELCOPYI** on an ISPF command line. Alternatively, SELCOPY/i may have been included as a selectable item in an ISPF menu and/or as an ISPF command (e.g. SI) by your systems programmer as part of the product install. ISPF screen management is used and so SELCOPY/i must have first been defined as an ISPF application. See *SELCOPY Products Suite Customisation Guide* for further information on configuring access to SELCOPY/i in ISPF.
	VTAM	Enter **LOGON APPLID(SELCOPYI)** on a VTAM USS screen. **Note:** The SELCOPY/i VTAM session controller program (CBLIVTAM) must be running and the SELCOPY/i VTAM applid, SELCOPYI, must be active. See *SELCOPY Products Suite Customisation Guide* for further information on configuring access to SELCOPY/i as a stand-alone VTAM application.
VSE	VTAM	Enter **LOGON APPLID(CBLIVTAM)** on a VTAM USS screen. Alternatively, SELCOPY/i may have been included as a selectable item in the VTAM Application Selection Menu, VTMUSSTR (SNA) or VTMUSSTB (non-SNA), by your systems programmer as part of the product install. **Note:** The SELCOPY/i VTAM session controller program (CBLIVTAM) must be running in a static or dynamic partition and the SELCOPY/i VTAM applid, CBLIVTAM, must be active. See *CBL Software Install Guide for VSE Systems* for further information.
VM	CMS	Enter **SELCOPYI** on a CMS command line to execute the SELCOPY/i startup Rexx EXEC.

Security Considerations

VSE Systems

By default, it is assumed that a Basic or Extended Security Manager (BSM or ESM) is operational (SEC=YES) and so SELCOPY/i authenticates the user logon id and password at startup, and thereafter performs resource access checking for the userid as required. e.g. For LIBR library lists and member edit.

Before an attempt is made to perform an operation on a resource, SELCOPY/i first checks whether the user has sufficient access authority for that resource and, if not, does not attempt the operation but instead returns an error message to the user's session.

When running with a security manager, SELCOPY/i processing of POWER queue list entries, displayed in the Execute POWER command window, is allowed only if the user's logon id matches either the TO or FROM field values.

If no security manager is in effect, then the SELCOPY/i System INI variable SYSTEM.VSESMLogon=No must be set in order to bypass userid authentication and resource access checking. The user is prompted for a logon id, which gets assigned to the environment variable %user%, but no password is required.

With no security manager, access to resources (LIBR libraries, members, etc.) will be unrestricted with the exception of POWER queue list entries where the user may process only queue entries that are password protected and for which the password is known to the user.

Whether or not a security manager is operational, SELCOPY/i may be customised to restrict its use to only a specified group of trusted users. Following logon, these users will be prompted for a SELCOPY/i password which may differ from the user's security manager password.

If the trusted user facility is activated, then no other users will be able to successfully start SELCOPY/i. Trusted users may process any POWER queue list entry without restriction.

Trusted users may process any POWER queue list entry without restriction.

See section *Security Manager* in document *CBL Software Install Guide for VSE Systems* for further information.

MVS Systems

On MVS systems, users login to SELCOPY/i using their RACF, or equivalent security package, login id. Under TSO, no SELCOPY/i login is performed as the user's TSO login id is used instead.

Thereafter, the user's access privilage is verified prior to performing any action on potentially protected resources. e.g. listing PDS(E) library contents and editing data sets.

In addition to this, the security administrator can restrict users' access to the following SELCOPY/i features using RACF profiles.

Resource Name	SELCOPY/i Feature
System	Access to the z/OS Operating System information available by selecting 'Operating System' from the Utilities/System menu in the CBLe main window menu bar, or via line commands `SYSI`, `SYSLPA`, `SYSLL`, `SYSAPF`, `SYSTASK`, `SYSSTOR`, `SYSPGM` and `CBLNAME`.
UserTSO	Log on to SELCOPY/i under TSO and ISPF.
UserVTAM	Log on to SELCOPY/i as VTAM application.
SELCOPY	Use of the SELCOPY Debug application.
CBLVCAT	Use of the CBLVCAT Interactive (VCI) application.
DB2	Use of DB2 features.

If these resources have been customised then users must have at least READ access to the specific resource in order to be able to perform the equivalent operation.

See section *Security Considerations* in document *SELCOPY Product Suite Customisation Guide* for further information.

3270 Terminal Emulation

Before SELCOPY/i is started, 3270 terminal session is required. Most installations now use a 3270 emulation software executing on a workstation rather than a real 3270 terminal.

3270 Screen Sizes

The CBL 3270 screen manager can operate in any 3270 screen size up to a total area of 16384 (16K) with a maximum width or depth of 255. This 16K area limit is imposed by the 14 bit address format of the 3270 data stream used by SELCOPY/i. The 255 width or depth limit is the result of some components using just 1 byte to store these dimensions.

All 3270 emulation software packages allow the user to configure a 3270 session to emulate hardware terminal models 2/3/4/5 having rows x columns screen sizes of 24x80, 32x80, 43x80 and 27x132 respectively.
Most good 3270 emulators also support the ability to define non-standard terminal sizes (dynamic TN3270 terminals) that allow users to obtain 3270 screen sizes with dimensions much larger than the standard hardware models.

Note: To configure z/OS non-standard screen sizes, a VTAM dynamic logmode must be defined to your system (IBM supply VTAM dynamic logmode D4C32XX3 in VTAMLIB.) This logmode may optionally be configured as the default for TN3270 sessions via a TELNETDEVICE DYNAMIC entry in the TN3270 server configuration data set.
See the IBM Technotes and Tips web page entitled *Creating dynamic 3270 screen size definitions for increased productivity* for further information.

When using 3270 emulation software, it is strongly recommended that the largest possible screen size be used to take full advantage of SELCOPY/i's ablility to display multiple overlapping windows. At CBL, a screen size of **96x160** is regularly used for TSO sessions (160 being the maximum number of columns supported by ISPF) and **86x190** used for CMS and VTAM sessions (MVS and VSE).

A selection of popular 3270 emulator packages have been installed and tested at CBL to determine support for dynamic terminal sizes and other features considered useful to SELCOPY/i operation (Keyboard macros, etc.) A synopsis of results for each product tested to date may be found at the *SELCOPY/i and 3270 Terminal Emulation Software* web page.

Sample 3270 emulator configuration files that provide non-standard terminal sizes have been generated for selected 3270 emulation products and are available for download from the *SELCOPY/i Downloads* web page.

Keyboard and Mouse Mapping

Configuration of the 3270 emulator software keyboard and mouse is recommended to greatly enhance the user's experience of the SELCOPY/i window environment. Most good 3270 emulator software packages provide the facility to do this.

Traditionally, 3270 terminal keyboards provide only the 24 programmable function keys (PF01-PF24) whereas 3270 emulation software provides the facility to map functions and macros to a much wider range of key combinations. (e.g. Ctrl-S may be mapped to execute a SAVE operation.)

Of most benefit to the use of SELCOPY/i, is the ability to map the mouse left button double-click action to be "position the cursor here and press <Enter>". Configuring this emulator feature provides quicker window manipulation, button pressing, menu selection, etc., so allowing the user to operate on SELCOPY/i windows using the mouse in a manner that is intuitive to users of PC or UNIX workstations.

In addition to sample 3270 session configuration files, CBL provides a library of keyboard macros with recommended keyboard and mouse map files for 3270 emulator products, IBM Personal Communications and Tom Brennan's Vista3270.

For further information, see the CBL web pages *SELCOPY/i and 3270 Terminal Emulation Software* and *SELCOPY/i Downloads*.

Window Concepts

The SELCOPY/i window environment is managed by the CBL3270 screen manager.

In general, CBL3270 managed windows behave in a similar fashion to window GUI environments provided by PC and UNIX operating systems.

This section provides technical detail on characteristics and concepts shared by all SELCOPY/i windows. It also provides instructions on how to work with windows in the SELCOPY/i environment.

Window Hierarchy

All windows exist in a hierarchy. At the top of the hierarchy is the desktop window which is automatically created during initialisation. The desktop window occupies the entire screen and cannot be moved, resized or destroyed. All other windows, including the SELCOPY/i main window, are dependents of the desktop.

When an application creates a window the new window has to be dependent on an existing window, the parent or owning window. If the application does not supply an existing window then the desktop window is used by default. This dependency relationship has two forms:

- Owned window. The dependent window is owned by the existing window which is called its owner. The owned window can be moved all over the display surface but is always in front of (cannot be obscured by) its owner. Typically owned windows are used for more complex entities such as MDI frame windows, dialog boxes and help windows.
- Child window. The dependent window is not only owned by, but is also a child of the existing window which is called its parent. The child window can only exist within the rectangle defined by its parent's client area. Typically, child windows are used for low level entities such as buttons and input fields.
- MDI (Multiple Document Interface) Child window. The dependent window is an MDI child of the existing MDI parent (frame) window. Like child windows, the MDI child window can only exist within the rectangle defined by its parent's MDI client area. However, unlike child windows, each MDI child window has a sizing border, a title bar, a window menu, a minimise button, and a maximise button.

  SELCOPY/i MDI applications include the CBLe text editor and SELCOPY Debugger, each supporting various types of MDI child windows. e.g. text edit views, SDE edit views, list windows, help windows and IPO panels.

When a window is destroyed, so are all of its dependent owned and child windows.

Manipulating Windows

Moving a window

If a window has a title bar it can be moved with the following procedure:

1. Place the cursor in the title bar of the window.
2. Press the <Enter> key. The window border will be highlighted.
3. Move the cursor to a new position.
4. Press the <Enter> key. The window will move by an amount equal to the displacement of the cursor.

Note that, if configured, double-clicking the left mouse button on the window title bar, then doing the same at the new location will perform the same action.

Also see line command MOVEWINDOW and, for CBLe text edit document windows only, SET WINPOS.

Resizing a window

If a window has a border it can be resized with the following procedure:

1. Place the cursor in the border of the window. If the cursor is in the top or bottom border, the window will have its depth changed. If the cursor is in the left or right border it will have its width changed. If the cursor is in a corner of the border it will have its width and depth changed.
2. Press the <Enter> key. The window border will be highlighted.
3. Move the cursor to a new position.
4. Press the <Enter> key. The window will be resized by an amount equal to the displacement of the cursor.

Note that, if configured, double-clicking the left mouse button on a window border, then doing the same at the new location will perform the same action.

Also see line commands SIZEWINDOW, DRAGBORDERMINUS, DRAGBORDERPLUS and, for CBLe text edit document windows only, SET WINSIZE.

Maximising a window

If a window has a maximise button it can be maximised by moving the cursor to the maximise button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the maximise button. The window will then take up the whole of the 3270 screen. The maximise button will change from a plus sign to a solid vertical bar (representing restore).

Also see line command MAXIMISE and, for CBLe text edit document windows only, WINDOW MAX.

Minimising a window

If a window has a minimise button it can be minimised by moving the cursor to the minimise button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the minimise button. The window will then be removed from the display and replaced by a small iconic window showing just a portion of its title bar near the botton of the SELCOPY/i main window. The minimise button will change from a minus sign to a solid vertical bar (representing restore).

Also see line command MINIMISE, for CBLe text edit document windows only, WINDOW MIN.

Restoring a window

When a window has been maximised or minimised, the maximise/minimise button is replaced restore button.

A maximised or minimised window may be restored to its former location and size by moving the cursor to the restore button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the restore button.

Also see line command RESTORE, for CBLe text edit document windows only, WINDOW REST.

Closing a window

A window can be closed by moving the cursor to the close button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the close button.

Also see line command CLOSE and, for CBLe text edit document windows only, WINDOW CLOSE.

Window Format

SELCOPY/i windows have a standard format which consists of the following components. Not all windows have all these components, but where present they occupy the same relative position in the window and have the same function.

The Title Bar
The title bar contains the title of the window.

The System Menu Button
The system menu button is at the left end of the title bar. If pressed the options of the system menu are displayed in a popup menu.

The Minimise Button
The minimise button is at the right end of the title bar. It is represented as a single "-" (minus sign) in white reverse video. If the window is minimised then the minus sign is replaced by a solid vertical bar representing restore.
Note that the Minimise button is not present in the SELCOPY/i main window.

The Restore Button
The Restore button is displayed in place of the Minimise or Maximise button when a window is in a minimised or maximised state respectively. It is represented by a solid vertical bar in white reverse video.
Note that the Restore button is not present in the SELCOPY/i main window.

The Maximise Button
The maximise button is at the right end of the title bar. It is represented as a single "+" (plus sign) in white reverse video. If the window is maximised then the plus sign is replaced by a solid vertical bar representing restore.
Note that the Maximise button is not present in the SELCOPY/i main window.

The Close Button

The close button is at the right end of the title bar. It is represented as an "x" in red reverse video.

The Menu Bar

The menu bar occupies one or more lines below the title bar. It contains optional items that can be selected by positioning the cursor on the menu item text and hitting <Enter> or, if configured, double-clicking the mouse left button on the menu item.

Menu items may display pop-up sub-menu items which may be selected in the same fashion. All menu items have the 3270 unprotected attribute so they can be navigated using the <Tab> key which tabs to enterable fields. Any data entered into a menu item is ignored.

Menu items may have been enabled or disabled by the SELCOPY/i application. When enabled they are displayed in white, when disabled they are displayed in blue in which case their selection will have no effect.

The Client Area

The client area of a window is the main body of the window and its contents vary depending on the class of the window.

The Command Line

The command line is an area of the window into which text commands may be entered. Most menu items have a command line equivalent text command.

Pressing Buttons

The 3270 architecture is such that the host is only informed of user input when a certain class of key is pressed on the keyboard. These keys are those with attention identifiers and typically consist of the function keys, the enter key, the Program Attention (PA) keys and a few other specialised keys. Even when operating under the control of a workstation 3270 emulator, the 3270 host application is not sensitive to mouse movements (except in the case when the emulator allows the user to assign a particular 3270 function to the mouse buttons).

Because of this, 3270 window buttons cannot be pressed in the same way as workstation window buttons. 3270 window buttons are pressed by positioning the cursor on the button and hitting <Enter> or, if configured, double-clicking the left mouse button on the button icon.

Window Focus

The focus window is the window which contains the cursor when the 3270 screen is displayed. If the window has a title bar then the fact that it is the focus window is indicated by colouring the title bar area with blue reverse video. All other windows have a white reverse video caption bar.

Input fields

The focus window defines the input rectangle which is the only area of the screen where input is enabled. When the focus window is not a child window the input rectangle is the window itself.

When the focus window is a child window the input rectangle is that defined by its parent. Any input fields outside of the input rectangle are temporarily disabled. Each field in the input rectangle can be visited by using the cursor tab key (shift+cursor tab key to reverse the direction).

Changing the focus window

The user can change the focus window in the following ways:

- By placing the cursor in a window and hitting <Enter> or, if configured, double-clicking the left mouse button in the window display area. This sets the focus to this window.
- By using the PREVWINDOW, NEXTWINDOW, MDIPREV or MDINEXT line commands. These commands cycle through all windows in creation sequence. By default, PF9 is assigned to MDINEXT in MDI applications such as CBLe and SELCOPY Debug, and NEXTWINDOW elsewhere.
- By selecting 'All Windows', from the Window menu in the CBLe main window menu bar, or executing the WINDOWLIST command to open the Window List window and then selecting the required window entry. The selected window becomes the focus window.
- By using the SETFOCUS line command to explicitly name the focus window.

Window Names

All the windows defined by a CBL3270 application have a name. The name is supplied by the application when the window is created and may be changed later during the window's life.

If the name is not supplied by the application then a default name is supplied by CBL3270 made up of the window class name suffixed with a three digit number which is incremented by 1 for each window of the class created during the CBL3270 session.

The main use of window names is to allow commands entered on the command line of a window to refer to other windows which are currently part of the application.

Viewing Window Names

The window name associated with each window in the SELCOPY/i session may be displayed in that window's title bar (and subsequently hidden from view) via the following:

- Select 'Display/Hide Window Names' from the system menu belonging to any open window.
- Enter the line command WINDOWNAMES on the command line of any window.

Either of these operations will display or hide the window names for **all** open windows in the SELCOPY/i session.

By default, display of window names is suppressed to avoid overcrowding the title bar.

Alternatively, the Window List window may be used to obtain a window's associated window name. The window list displays all open windows and their associated window names (including MDI window names.)

Window Class

All the windows defined by a CBL3270 application are members of a window class.

A window class is identified by a 1 to 8 character name and defines window behaviour, appearance and functions that are commmon to windows belonging to that window class.

CBL3270 uses the window class name to associate a processing module (called a Window Procedure) with a window.

All windows in the same class are managed by the same window procedure. The window procedure is called by CBL3270 whenever an event occurs which affects the window. It is the window procedure's responsibility to:

1. Paint the window client area.
2. Establish PFKey definitions for the individual Class Function Keys table.
3. Process any data entered into the window.
4. Respond to any commands issued from the window's menu or command line.

A complete list of SELCOPY/i Windows Classes may be found in *Appendix A - SELCOPY/i Window Classes*.

System Menu

The system menu is a menu of functions which is available on all windows within a CBL3270 environment.

You access the system menu by pressing the system menu button at the top left of the main window (or any subordinate window that has a system menu button). The following options are available from the system menu:

Layout	For Storage Display Windows only, displays the options popup menu. (As for SHOWPOPUPMENU.)
Restore	Restore a maximised or miminised window to its original size and position.
Minimise	Minimise the window with the system menu.
Maximise	Maximise the window with the system menu.
Close	Close the window with the system menu.
Quit	Quit the application.
Window List	Open a window showing the current list of windows. You can select a window from this list by placing the cursor on a list element and hitting <Enter> or, if configured, double-clicking the left mouse button. The selected window will be given the focus.
Next window	Give the focus to the next window in the hierarchy of open windows.
Previous window	Give the focus to the previous window in the hierarchy of open windows.
Command line	Open the command line dialog.
Function keys	Open the functions keys dialog.
Show/Hide window names	Toggle the status of the window names display. Window names are unique window identifiers which can be used in commands to identify a particular window. If displayed, the window name is displayed in the title bar as a prefix to the window caption.
Use ISPF/TSO	When running under TSO/ISPF, this menu command toggles the display from ISPF to TSO format.

Function Keys

Function key command tables are maintained for keys PF01 to PF24 at the following levels:

Window	For each window instance.
Class	For each window class.
Caption	One table for the caption area (title bar) of all windows.
Border	One table for the border area of all windows.
Default	One table of system defaults.

These function key levels correspond function key command tables that are searched in the following order until a command is found that is assigned to the function key pressed:

1. If cursor is positioned in a window's caption area, search the Caption function key table.
2. If cursor is positioned on a window's border, search the Border function key table.
3. The Window function key table belonging to the focus window.
4. The Class function key table for the Window Class associated with the focus window. e.g. EDTWEDIT for CBLe text edit document windows, SDEWVIEW for CBLe SDE document windows, etc.
5. The Default function key table.
6. Repeat the search for function tables belonging to the parent window of the focus window.

If no command assignment is found for the function key, then no command is issued.

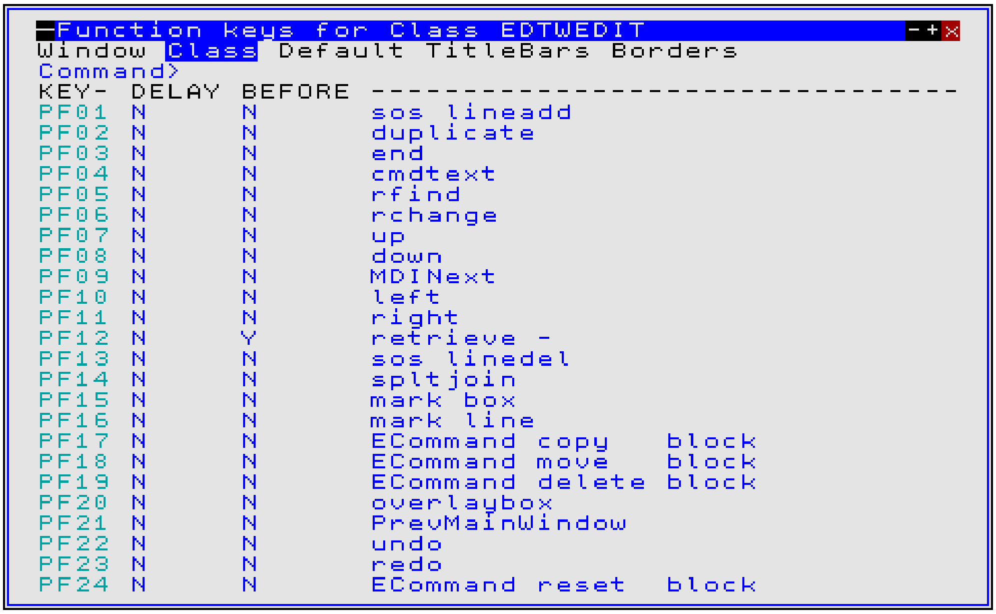
Function keys settings are displayed and updated using the Function Keys window which may be opened via the following:

- Select 'Function Keys' from the System menu belonging to the window for which function keys are to be displayed.
- Enter the command KEYS on the command line of the window for which function keys are to be displayed.

The Function Keys window displays the current PFKey table settings for a CBL3270 window.

The function key table for any of the levels may be displayed and updated by first selecting the required level (Window, Class, Default, TitleBars, Borders) from the Function Keys menu.

All the displayed fields, other than the Key field, are enterable. Changes made to the table are only implemented once the user confirms the changes on exit of the Function Keys window.



```
Function keys for Class EDTWEDIT
Window Class Default TitleBars Borders
Command>
KEY- DELAY BEFORE --------------------------------
PF01 N     N      sos lineadd
PF02 N     N      duplicate
PF03 N     N      end
PF04 N     N      cmdtext
PF05 N     N      rfind
PF06 N     N      rchange
PF07 N     N      up
PF08 N     N      down
PF09 N     N      MDINext
PF10 N     N      left
PF11 N     N      right
PF12 N     Y      retrieve -
PF13 N     N      sos linedel
PF14 N     N      spltjoin
PF15 N     N      mark box
PF16 N     N      mark line
PF17 N     N      ECommand copy   block
PF18 N     N      ECommand move   block
PF19 N     N      ECommand delete block
PF20 N     N      overlaybox
PF21 N     N      PrevMainWindow
PF22 N     N      undo
PF23 N     N      redo
PF24 N     N      ECommand reset  block
```

Figure 1. Function Keys Window.

Columns Displayed

Name	Description
KEY	PFKeys 01-24 (Non-enterable field)
DELAY	Determine whether the associated command is executed immediately when the function key is hit or merely placed on the local command line. Valid entries are Y or N.

BEFORE	Determine whether the associated command is executed before or after any other CBL3270 screen input. (e.g. a command line command or prefix area command.) Valid entries are Y or N.
untitled	The command(s) associated with the PFKey. For CBLe, a number of commands may be issued if separated by an appropriate separator character.

Default Function Keys

SELCOPY/i is distributed with the following PFKey assignments in the **Default** function key table:

PF1	Top	Display data at the top of a scrollable window.
PF2	Bottom	Display data at the bottom of a scrollable window.
PF3	Close	Close the window in which the cursor is positioned. (N.B. not necessarily the current focus window)
PF4	CmdText	Issue command at the cursor position if prefixed by "<" (less than), otherwise place text on the command line.
PF7	Up	Scroll up the file so that the current focus line becomes the last line of the display.
PF8	Down	Scroll down the file so that the current focus line becomes the first line of the display.
PF9	NextMainWindow	Place focus on the next main window in the SELCOPY/i window list.
PF10	Left	Scroll the display to the left so that the current focus column becomes the last column of the display.
PF11	Right	Scroll the display to the right so that the current focus column becomes the first column of the display.
PF12	Retrieve -	Retrieve the last command issued and place it at the command line.
PF13	ShowPopupMenu	For storage windows only, display the options popup menu.
PF16	CmdText Edit	Issue command at the cursor position if **not** prefixed by "<" (less than), otherwise place text on the command line.
PF21	PrevMainWindow	Place focus on the previous main window in the SELCOPY/i window list.
PF24	Retrieve +	Retrieve the next command issued and place it at the command line.

SELCOPY/i is distributed with the following PFKey assignments in the **Caption** (title bar) function key table:

PF7	MoveWindow by y=-1	Reposition window up 1 line
PF8	MoveWindow by y=+1	Reposition window down 1 line.
PF10	MoveWindow by x=-1	Reposition window left 1 column.
PF11	MoveWindow by x=+1	Reposition window right 1 column.

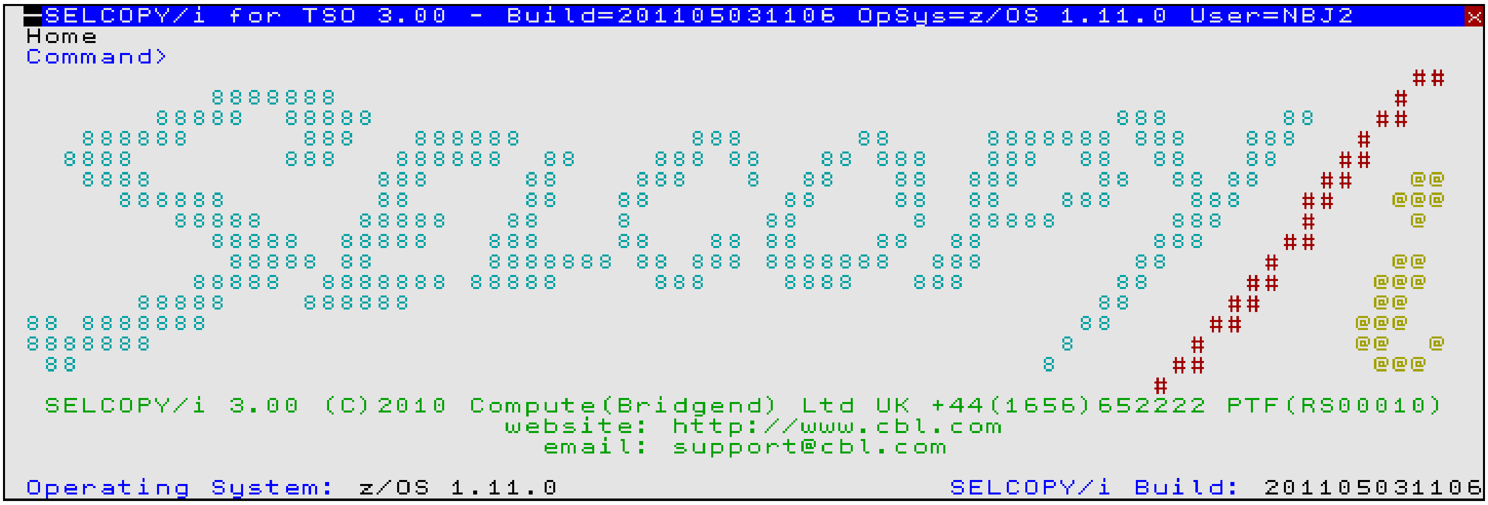
SELCOPY/i is distributed with the following PFKey assignments in the **Borders** function key table:

PF7	DragBorderMinus	Move the border on which the cursor is position 1 column and/or row towards the top left corner of the 3270 display.
PF8	DragBorderPlus	Move the border on which the cursor is position 1 column and/or row away from the top left corner of the 3270 display.
PF10	DragBorderMinus	Same as PF7.
PF11	DragBorderPlus	Same as PF8.

SELCOPY/i Main Window

On starting SELCOPY/i the main window, in which all SELCOPY/i applications execute, is displayed in a maximised state. Note that this window cannot be resized.

Since the CBLe text editor application is executed automatically on SELCOPY/i startup and the CBLe main window is also opened in a maximised state, the SELCOPY/i main window is not normally visible until all CBLe text editor windows are closed.



```
SELCOPY/i for TSO 3.00 - Build=201105031106 OpSys=z/OS 1.11.0 User=NBJ2
Home
Command>

 SELCOPY/i 3.00 (C)2010 Compute(Bridgend) Ltd UK +44(1656)652222 PTF(RS00010)
                         website: http://www.cbl.com
                           email: support@cbl.com

Operating System: z/OS 1.11.0                       SELCOPY/i Build: 201105031106
```

Figure 2. SELCOPY/i Main Window.

The main window contains:

- A title bar. Located at the top of the window, it includes, from left to right:
    - ♦ A system menu button at the extreme left. This button accesses the system menu options.
    - ♦ If WINDOWNAMES is active, the name of the window (VCIWMAIN).
    - ♦ The SELCOPY/i product name, operating environment, release and build level.
    - ♦ The operating system release.
    - ♦ The user's logon id.
    - ♦ A close button as the first character from the right.
- The menu bar. Located immediately below the title bar, it lists the SELCOPY/i main window menu bar items.
- The client area. Occupying the body of the window, it contains:
    - ♦ The SELCOPY/i logo, release, copyright notice and CBL contact details.
    - ♦ The operating system name and version.
    - ♦ The user id.
    - ♦ The build level of SELCOPY/i. This information is useful when raising a product query with CBL.
- The command line. This may be positioned at either the top or bottom of the main window.
  Commands may be entered at the **Command>** prompt to invoke SELCOPY/i facilities that duplicate or extend the menu facilities.

You can launch SELCOPY/i facilities by doing the following:

- Select a SELCOPY/i main menu bar item by positioning the cursor on the menu item text and hitting <Enter> or, if configured, double-clicking the left mouse button on the menu item.
- Executing a line command.

SELCOPY/i Main Window Menu Bar

The SELCOPY/i main menu bar is located at the top of the SELCOPY/i main window.

The main menu consists of the following item:

Home

Open the CBLe Text Editor application and edit the user's HOME command centre file. This is equivalent to executing the line command HOME.

All SELCOPY/i facilities may be started from the CBLe Text Editor application and run as a child window of the CBLe parent window. The exception to this is the SELCOPY Debug application which, although started from CBLe, operates as a separate, special instance of the CBLe text editor. In some cases, SELCOPY/i facilities require a CBLe text edit environment to operate successfully. e.g. Structured Edit (SDE), DB2, File Search, Update & Copy.

Because of this, menu items for all available SELCOPY/i facilities are located in the CBLe main window menu bar and not the SELCOPY/i main window menu bar.

SELCOPY/i Clipboard

SELCOPY/i supports a clipboard facility to allow users to Copy, Cut and Paste data between windows running in the SELCOPY/i environment that support clip board functions.

At this time, clipboard facilities are only supported in CBLe and SDE edit and browse views. e.g. Data copied to the clipboard from an edit view in the SELCOPY Debug application may be pasted to an edit view in the CBLe text edit application.

Note that the SELCOPY/i clipboard is not associated with any other clipboard facility offered by the system.

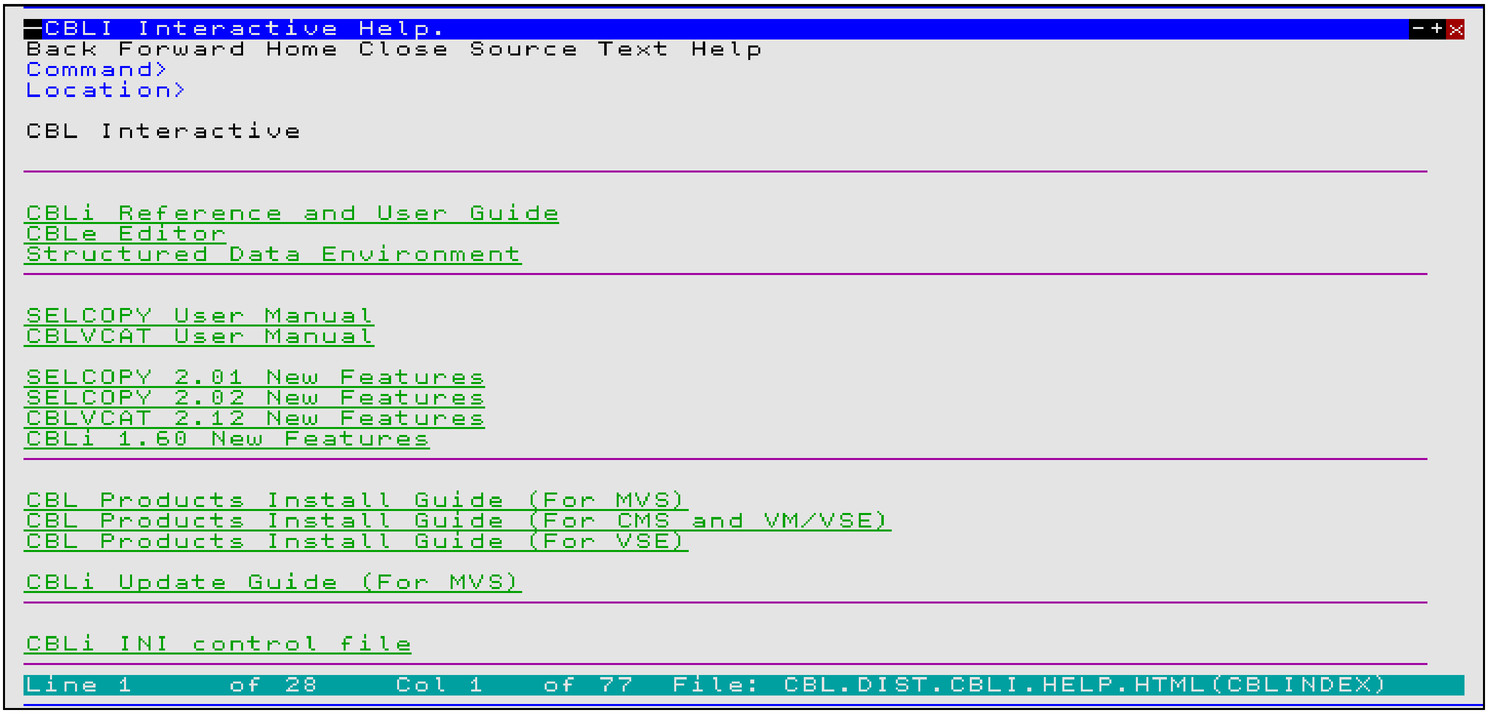
SELCOPY/i Interactive Help

The SELCOPY/i Interactive Help windows are basic HTML browse windows started with links to the SELCOPY/i suite of HTML help files located via the SELCOPY/i INI variable Help.DefaultPath.

SELCOPY/i Help may be obtained via the following:

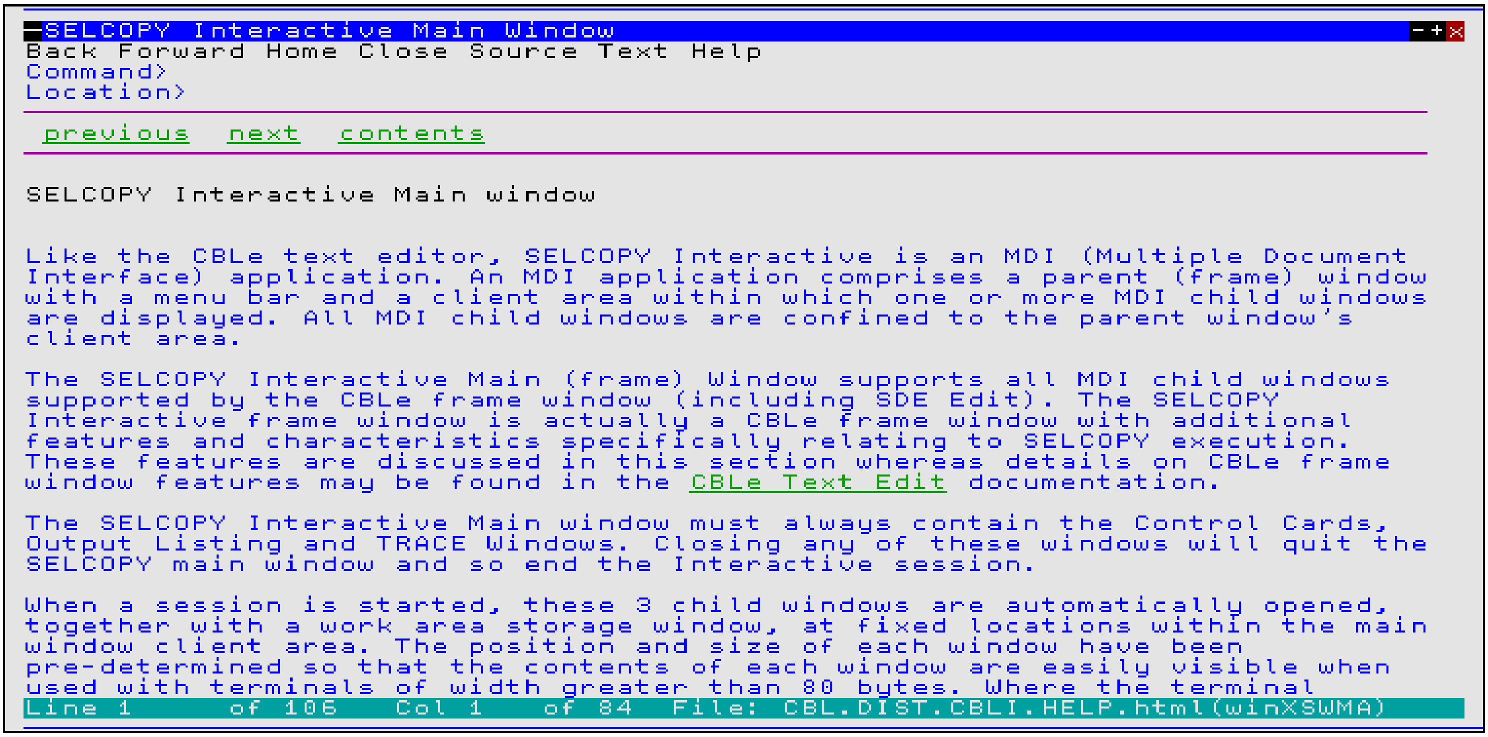
- Select 'Help' from the the CBLe main window menu bar.
- Enter command HELP on the command line of any window.

Context sensitive help windows are displayed by performing either of these functions within the appropriate SELCOPY/i window. e.g. Executing HELP in a Library List window opens the help window for the topic List Library Members.



```
CBLI Interactive Help.
Back Forward Home Close Source Text Help
Command>
Location>

CBL Interactive

CBLi Reference and User Guide
CBLe Editor
Structured Data Environment

SELCOPY User Manual
CBLVCAT User Manual

SELCOPY 2.01 New Features
SELCOPY 2.02 New Features
CBLVCAT 2.12 New Features
CBLi 1.60 New Features

CBL Products Install Guide (For MVS)
CBL Products Install Guide (For CMS and VM/VSE)
CBL Products Install Guide (For VSE)

CBLi Update Guide (For MVS)

CBLi INI control file
Line 1      of 28      Col 1     of 77    File: CBL.DIST.CBLI.HELP.HTML(CBLINDEX)
```

Figure 3. SELCOPY/i Main Help Menu window.



```
 SELCOPY Interactive Main Window                                          -+x
Back Forward Home Close Source Text Help
Command>
Location>

 previous  next  contents

SELCOPY Interactive Main window

Like the CBLe text editor, SELCOPY Interactive is an MDI (Multiple Document
Interface) application. An MDI application comprises a parent (frame) window
with a menu bar and a client area within which one or more MDI child windows
are displayed. All MDI child windows are confined to the parent window's
client area.

The SELCOPY Interactive Main (frame) Window supports all MDI child windows
supported by the CBLe frame window (including SDE Edit). The SELCOPY
Interactive frame window is actually a CBLe frame window with additional
features and characteristics specifically relating to SELCOPY execution.
These features are discussed in this section whereas details on CBLe frame
window features may be found in the CBLe Text Edit documentation.

The SELCOPY Interactive Main window must always contain the Control Cards,
Output Listing and TRACE Windows. Closing any of these windows will quit the
SELCOPY main window and so end the Interactive session.

When a session is started, these 3 child windows are automatically opened,
together with a work area storage window, at fixed locations within the main
window client area. The position and size of each window have been
pre-determined so that the contents of each window are easily visible when
used with terminals of width greater than 80 bytes. Where the terminal
Line 1      of 106    Col 1    of 84   File: CBL.DIST.CBLI.HELP.html(winXSWMA)
```

Figure 4. Help window for SELCOPY Debug.

`Back`
: Display the HTML page that occurs immediately prior to the current page in the stack of viewed pages.

`Forward`
: Display the HTML page that occurs immediately after the current page in the stack of viewed pages.

`Home`
: Display the defined Home page.

`Close`
: Close the current HTML browse window.

`Refresh`
: Refresh (reload) the current HTML page.

`Find`
: Open a dialog window to locate lines in the current HTML page that contain a specified string.
Not yet supported.

`Source`
: Open a CBLe text editor window to edit the source for the current HTML page.

`Options`
: Tailor options for the current HTML browse window.
Not yet supported.

`Text`
: Open a CBLe text editor window to edit the current HTML page as plain text with file name UNTITLED.

`Help`
: Display this help page.

`Location>`
: Specify an explicit HTML source fileid. If only a file name is specified, then the HTML browse window will search the Help.Defaultpath library for that file name.

This allows the user to display any file containing basic HTML tags that is not necessarily associated with the SELCOPY/i suite of help files.

Window Classes

Every SELCOPY/i window belongs to a named window class. Window classes define a set of window characteristics that are common to all windows assigned the same window class.

These characteristics include the window's appearance such as the existance of a command prompt, menu bar, scroll field, input fields; if present, the contents of the menu bar; PFKey assignments and the presentation format of data to be displayed in a window's display area.

A comprehensive list of SELCOPY/i window classes may be found in *Appendix A - SELCOPY/i Window Classes*.

Characteristics of CBLe Text Edit window classes, including Structured Data Edit document views, are detailed in the *SELCOPY/i Text Editor* documentation.

Many window classes are one-to-one with an individual SELCOPY/i feature and their characteristics are documented with that particular feature. (e.g. window class EDTWSORT is specific to the Text Edit SORT dialog window.) Characteristics and behaviour of those window classes that are common to more than one SELCOPY/i feature are documented in this chapter.

Storage Display Windows

Storage display windows (window class HEXDUMP, STORAGE and EDTWHEXE) provide a view of an area of storage in dump format.

SELCOPY/i windows that utilise storage display windows are:

- CBLNAME display window.
- SELCOPY Debug Workarea windows.
- SELCOPY Debug POS windows.
- CBLe text edit line HEX dump views.

Storage Window Display Format

The length of storage data displayed may be restricted by the type of window opened. e.g. The amount of data displayed in a CBLe HEX window is restricted to be the length of the focus line.

Each row of a storage display window has the following format:

Field Width	Type	Description
8	Hex	Address in storage of the displayed data.
6	UInt	Displacement from the start address of the displayed data.
8,16,32 or 64	Hex	Data in storage in hexadecimal format. (4, 8, 16 or 32 bytes depending upon window size).
4,8 ,16 or 32	Char	Data in storage in character format. (Field width adjusts automatically to match hexadecimal display width.)

The format of the display may be updated using the options popup menu which may be opened using the SHOWPOPUPMENU command or via the system menu button of the storage display window. By default, the SHOWPOPUPMENU command is assigned to PF5 in storage display windows.

Storage Window Resizing

User resizing of a storage display window's width will always be adjusted by SELCOPY/i to display one of the valid data display widths. i.e. 1, 2, 4 or 8 words of hexadecimal data plus its equivalent character representation if required.

If the window's width is increased or decreased by one column, the window's width is rounded up or down respectively to equal the next valid display width. e.g. When using DRAGBORDERPLUS and DRAGBORDERMINUS (assigned to PF8/PF11 and PF7/PF10 respectively) on either of the window's vertical borders.
If the window's width is increased or decreased by more than one column, the window's width is always rounded down to equal the next valid display width.

Storage Display Navigation

The displacement field in the first row of a storage display window is an enterable field (highlighted in red by default.) This field may be overwritten with a displacement, from the start address, of the byte that should be displayed first in the storage window's display area.

Line commands UP CURSOR and DOWN CURSOR may be used to navigate the storage display window. By default, UP CURSOR is assigned to PF7 and DOWN CURSOR is assigned to PF8.

Storage Data Manipulation

Some storage display window invocations allow the data to be updated simply by overtyping the existing character or hexadecimal representation and then, to commit the change, hitting <Enter>.

Only areas of storage that are not write protected may be altered. Beware when altering storage as this could adversely affect programs that utilise the updated area of storage.

List Windows

A list window is a child window with a parent window class of LISTFRAM, LISTFILE, VCIWEXEC or WINWIPO0.

List windows display information as a table of rows and columns. The content of the list columns is described by data elements, each having a name and data type, and defined by a Field Descriptor Block (FDB).

SELCOPY/i windows that have child list windows include DB2 Object lists, DASD lists, Dataset lists, Execute CBLVCAT and Execute IDCAMS windows, APF Authorised Library lists and FDB lists.

In addition to any features defined by a list window's parent window class, list windows provide support for features described in this section.

- List Window Status Bar
- List Window Menu
- Selecting, Sorting and Filtering
- Sorting with the Cursor
- List Entry Location
- List Window Prefix Area

List Window Status Bar

All list windows have a status bar that occupies the last row of the window display area.

The list window status bar displays the following information:

Status bar display	Description
Line *c_row* of *n_rows*	Identifies the list current row value (*c_row*) and the total number of rows in the list (*n_rows*).
Col *c_col* of *n_cols*	Identifies the list current column value (*c_col*) and the total number of list columns, including non-scrollable list columns, in the list (*n_cols*).
Views *n_views*	Identifies the number of saved list views (as described by SELECT, WHERE, SORT clause combinations) that exist for the current list data. (*n_views*)
select_clause <where_clause <sort_clause> >	Identifies the current view of the list data.

List Window Menu

All list windows have the following menu items:

View	This is a popup menu which lists all the views which have been made of the current list. You can select a view from this menu.
Back	Select the previous view. This is equivalent to executing command **BAck** in the current list window.
Forward	Select the next view. This is equivalent to executing command **FOrward** in the current list window.
FDB	Display the Field Descriptor Block for the list. This is equivalent to executing command **FDB** in the current list window.
Text	Open a CBLe text edit window containing a text version of the list. This is equivalent to executing command **TEXT** in the current list window.

Refresh	Refresh the contents of the list window so that all column fields reflect the current status. This is equivalent to executing command **REFRESH** in the current list window.
Help	Open the help window for the list. This is equivalent to executing command **HELP** in the current list window.

View List Display

A SELECT clause and/or WHERE clause, executed as a CLI command, creates a new view of the list data. On each execution of one of these CLI commands, the command stream is recorded as a single entry at the end of the list window's View List Display.

This allows the user to select and filter list columns and rows and then easily recall any previous view of the data.

The View List Display is a drop down menu available by selecting the "View" List menu bar item. Any previous view may be selected by positioning the cursor on the required SELECT and/or WHERE clause entry and hitting <Enter> or, if configured, double-click the left mouse button on the entry.

Alternatively, the display's view of the data may be switched to view immediately prior to or following the current view by selecting "Back" or "Forward" respectively from the List menu bar.

Field Descriptor Block (FDB)

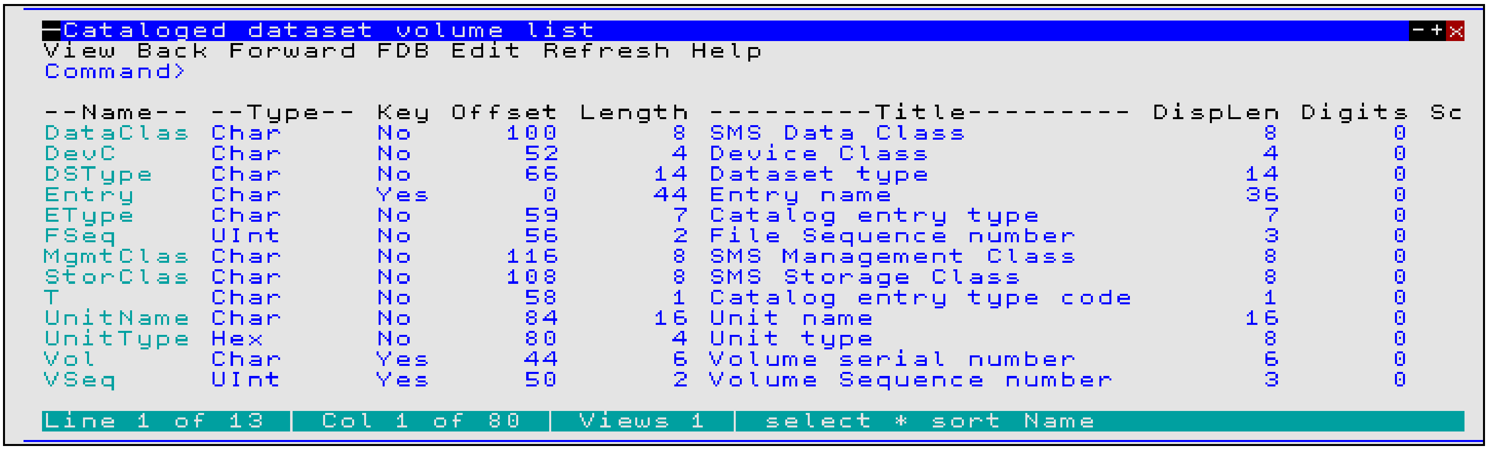
The FDB window may be opened using the following:

- Select 'FDB' from the list window menu bar.
- Enter command **FDB** on the list window command line.

Information about the data displayed under each column of a List window is referenced via an internal SELCOPY/i data structure. This structure includes, or addresses, fields that define the column data attributes. e.g. column name, column data type, column data length, etc.

The Field Descriptor Block (FDB) maps this internal structure and so provides information for all fields in the List window.

FDB is primarily used as an aid to performing List window SELECT, SORT and WHERE clause commands.



```
 Cataloged dataset volume list                                                 -+x
View Back Forward FDB Edit Refresh Help
Command>

--Name-- --Type-- Key Offset Length ---------Title--------- DispLen Digits Sc
DataClas Char     No     100      8 SMS Data Class                8      0
DevC     Char     No      52      4 Device Class                  4      0
DSType   Char     No      66     14 Dataset type                 14      0
Entry    Char     Yes      0     44 Entry name                   36      0
EType    Char     No      59      7 Catalog entry type            7      0
FSeq     UInt     No      56      2 File Sequence number          3      0
MgmtClas Char     No     116      8 SMS Management Class          8      0
StorClas Char     No     108      8 SMS Storage Class             8      0
T        Char     No      58      1 Catalog entry type code       1      0
UnitName Char     No      84     16 Unit name                    16      0
UnitType Hex      No      80      4 Unit type                     8      0
Vol      Char     Yes     44      6 Volume serial number          6      0
VSeq     UInt     Yes     50      2 Volume Sequence number        3      0

Line 1 of 13 | Col 1 of 80 | Views 1 | select * sort Name
```

Figure 5. FDB for Catalog List window.

Name

Specifies the field names that constitute the List window column headers. These entries are used when selecting columns and sorting/filtering rows to generate new list views.

Type

The data format in which data for that column is stored.

Key

Identifies whether or not the column is a key column.

If the column is a key column and is either the first column in the list or immediately follows another key column, then it is always in view even when scrolling the list view left or right.

Furthermore, if a key column selected as the first column of a list contains duplicate entries, then, when these entries are sorted together, only the first, in view occurrence of the key field value will be displayed. Subsequent list rows containing the duplicate key field values are displayed with ditto/quotation marks (") in the key field. If sorting on a different column separates these duplicate key field values so that they no longer appear on consecutive list rows, then the field values are once again displayed replacing the ditto mark. The display of values in all other columns are unaffected by this list window feature.

Offset

The offset within the structure of the column data.

Length

The length of the column data field within the structure. Note that if one or more levels of indirection to the column data field exist, then the structure contains an address field length 4.

Title

Descriptive name for the field column.

Displen

The length of the longest entry displayed in the column.

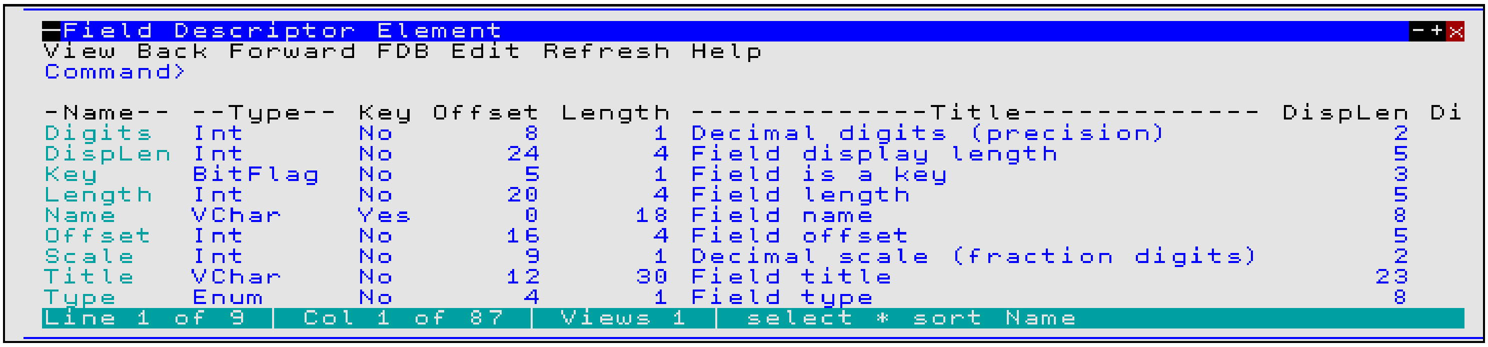
Digits

The number of decimal digits (precision) displayed for column data of numerical data type.

Scale

The number of scale digits (fraction digits) displayed for column data of numerical data type.

Because the FDB window is itself a List window with its column data attributes referenced by an internal data structure, it too may be described by an FDB and is subject to all supported list window functions (SELECT, SORT WHERE, etc.)



```
Field Descriptor Element                                                    -+x
View Back Forward FDB Edit Refresh Help
Command>

-Name-- --Type-- Key Offset Length ------------Title------------- DispLen Di
Digits   Int      No       8      1 Decimal digits (precision)          2
DispLen  Int      No      24      4 Field display length                5
Key      BitFlag  No       5      1 Field is a key                      3
Length   Int      No      20      4 Field length                        5
Name     VChar    Yes      0     18 Field name                          8
Offset   Int      No      16      4 Field offset                        5
Scale    Int      No       9      1 Decimal scale (fraction digits)     2
Title    VChar    No      12     30 Field title                        23
Type     Enum     No       4      1 Field type                          8
Line 1 of 9 | Col 1 of 87 | Views 1 | select * sort Name
```

Figure 6. FDB for FDB window.

Edit View

The contents of the current list view may be edited and saved to disk.

- Select 'Text' from the list window menu bar.
- Enter command **TEXT** on the list window command line.

A CBLe text editor window is opened, the list is loaded into CBLe storage and edited. The edited view is given a generic title "UNTITLED". On saving the text for the first time, the user is prompted to provide a valid "Save as" fileid.

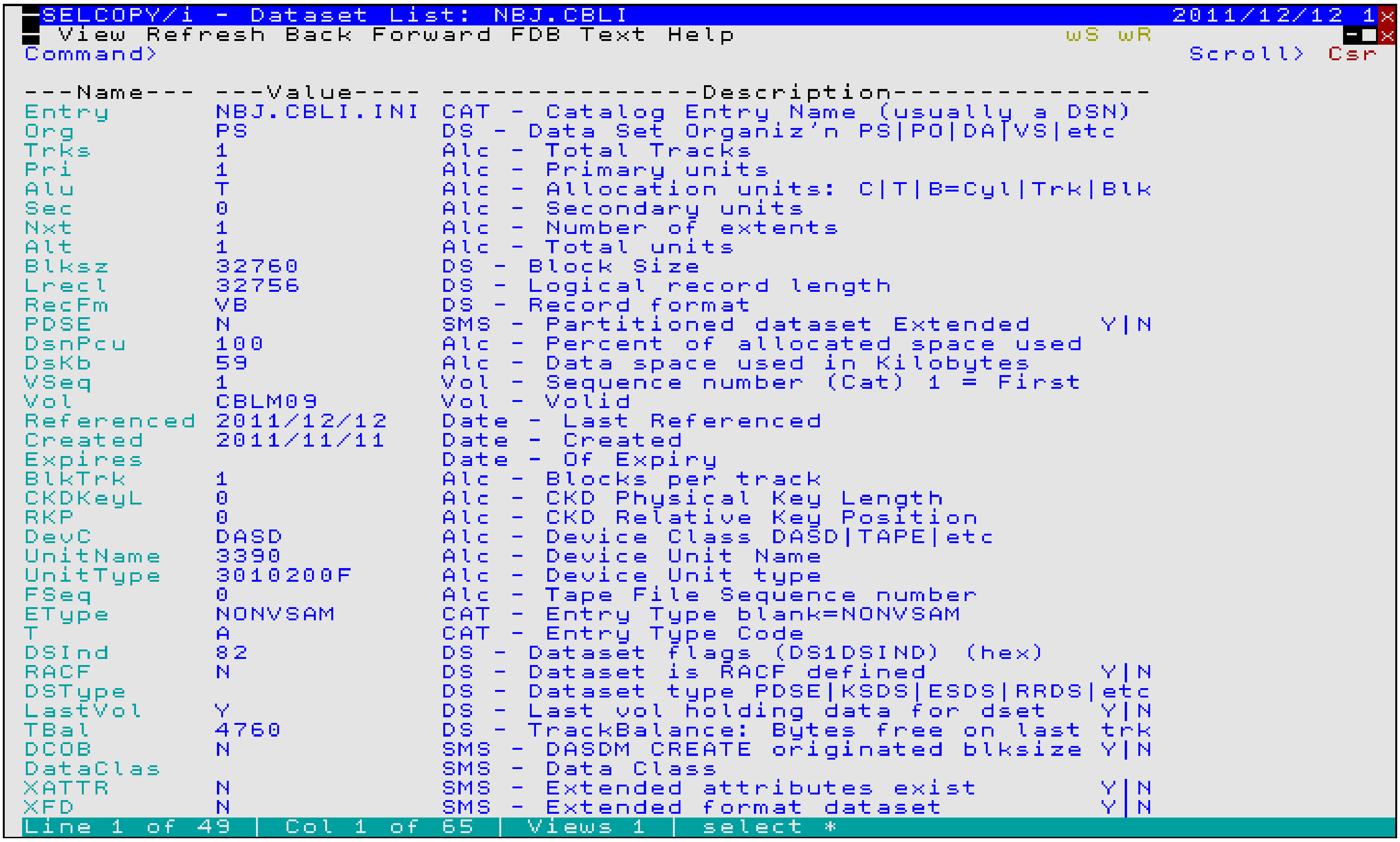
Note that if INSTANCE=SINGLE is set in the (Edit) section of the SELCOPY/i INI file and a CBLe editor window is already open, then the list will be edited in a new document window of the existing CBLe window.

Zoom View

The contents of a list entry may be displayed vertically in a single entry, zoomed view format. i.e each column value belonging to the selected entry is displayed in a separate row of the zoomed window view.

- Enter prefix command **>** against the required list window entry.

The zoomed window view is a separate list window which displays the selected list entry's field names, values are field descriptions (as reported in the FDB list window view.)



```
 SELCOPY/i - Dataset List: NBJ.CBLI                               2011/12/12 1
  View Refresh Back Forward FDB Text Help                   wS wR
Command>                                                           Scroll> Csr

---Name--- ---Value---- ---------------Description----------------
Entry      NBJ.CBLI.INI CAT - Catalog Entry Name (usually a DSN)
Org        PS           DS - Data Set Organiz'n PS|PO|DA|VS|etc
Trks       1            Alc - Total Tracks
Pri        1            Alc - Primary units
Alu        T            Alc - Allocation units: C|T|B=Cyl|Trk|Blk
Sec        0            Alc - Secondary units
Nxt        1            Alc - Number of extents
Alt        1            Alc - Total units
Blksz      32760        DS - Block Size
Lrecl      32756        DS - Logical record length
RecFm      VB           DS - Record format
PDSE       N            SMS - Partitioned dataset Extended    Y|N
DsnPcu     100          Alc - Percent of allocated space used
DsKb       59           Alc - Data space used in Kilobytes
VSeq       1            Vol - Sequence number (Cat) 1 = First
Vol        CBLM09       Vol - Volid
Referenced 2011/12/12   Date - Last Referenced
Created    2011/11/11   Date - Created
Expires                 Date - Of Expiry
BlkTrk     1            Alc - Blocks per track
CKDKeyL    0            Alc - CKD Physical Key Length
RKP        0            Alc - CKD Relative Key Position
DevC       DASD         Alc - Device Class DASD|TAPE|etc
UnitName   3390         Alc - Device Unit Name
UnitType   3010200F     Alc - Device Unit type
FSeq       0            Alc - Tape File Sequence number
EType      NONVSAM      CAT - Entry Type blank=NONVSAM
T          A            CAT - Entry Type Code
DSInd      82           DS - Dataset flags (DS1DSIND) (hex)
RACF       N            DS - Dataset is RACF defined          Y|N
DSType                  DS - Dataset type PDSE|KSDS|ESDS|RRDS|etc
LastVol    Y            DS - Last vol holding data for dset   Y|N
TBal       4760         DS - TrackBalance: Bytes free on last trk
DCOB       N            SMS - DASDM CREATE originated blksize Y|N
DataClas                SMS - Data Class
XATTR      N            SMS - Extended attributes exist       Y|N
XFD        N            SMS - Extended format dataset         Y|N
Line 1 of 49 | Col 1 of 65 | Views 1 | select *
```

Figure 7. Zoom Window View.

Columns Displayed

Name	Type	Description
Name	ALPair	Field name
Value	ALPair	Field Value
Description	ALPair	Field description

Selecting, Sorting and Filtering

Each list has a basic set of column data which is defined by the Field Descriptor Block (FDB) associated with the list. You can view the list of columns by selecting the FDB menu option or entering the FDB command on the list window command line.

You can modify the display by selecting your own list of columns from this basic list, specifying a filter to restrict the rows displayed and specifying a sort order. You do this by entering the select command on the list window command line. The select command syntax is similar to the SQL SELECT statement. It consists of one or more of the following:

- A SELECT clause which determines the columns displayed.
- A WHERE clause which filters the rows displayed by imposing a condition on the values in one or more columns.
- A SORT clause (ORDER BY clause) which displays the rows in an order determined by the values in one or more columns. (See also Sorting with the Cursor.)

Unlike SQL, these clauses can be given in any order, and any of them can be ommitted. The general syntax is:

```
>>--+-------------------+--+------------------+--+-----------------+-------><
    |                   |  |                  |  |                 |
    +-- select_clause --+  +-- where_clause --+  +-- sort_clause --+
```

Each time you issue a select command with a select or where clause you create a new view of the list. If you issue only a sort clause no new view is created, but the sort order is changed for the current view.

SELECT, WHERE and SORT clauses are not cumulative. Therefore, their execution replace any previously executed clause of the same type. e.g. Execution of a WHERE clause will not perform a logical AND with any previously executed WHERE clause, but will replace it instead. In addition to this, the clauses are hierarchical so that execution of a SELECT clause will undo any active WHERE and/or SORT clause, execution of a WHERE clause will undo any active SORT clause.

The set of views which you have is listed in the View menu option, from which you can select a view from the ones you have created with the select command.

SELECT Clause

Syntax:

```
               +------- ALL ----------------------------+
               |                                        |
>>-- SELect ---+-------------------+--+-----------------+-------------------->
               |                   |  |                 |
               |  +----- , -----+  |  | +-- , --+       |
               |  v             |  |  | |       |       |
               +--+- columname -+--+  +-+-------+-- * --+

 >-------------+------------------+---+------------------+-------------------><
               |                  |   |                  |
               +-| WHERE Clause |-+   +- | SORT Clause |-+
```

Description:

Specified as a CLI command or as a parameter on WHERE or SORT, the SELECT clause is used to identify field columns for display and the order in which they appear.

Use of a SELECT clause is not cumulative and so will replace those columns currently selected for display. It also resets any prevailing WHERE and/or SORT clause specifications.

Note that the last execution of a SELECT clause CLI command is stored and applied to any new List window of the **same type** (e.g. Library List window). This occurs whether the list window is opened within the same SELCOPY/i session or opened after the current SELCOPY/i session is ended and a new session started.

Executing a SELECT clause will add a new entry to the View List Display drop down menu.

Parameters:

`ALL`

ALL returns all columns.

`columname`

Name of the list window column to select.

Multiple column names must be separated by commas and/or one or more blanks. Column names may be specified in any order and any number of times.

`*`

Display all remaining column names that have not already been selected, in their default order of display.
This is the same as ALL if no other column names are specified.

`WHERE Clause`

Any valid WHERE clause used to filter list rows.

`SORT Clause`

Any valid SORT clause used to sort list rows.

The Where Clause

Syntax:

```
                                       +----+-- AND --+----+
                                       |    |         |    |
                                       |    +-- OR ---+    |
                                       v                   |
>>-- WHere ---+---------+---+-------+-----| Filter_Expr |--+--+-------+------->
              |         |   |       |                         |       |
              +-- NOT --+   +-- ( --+                         +-- ) --+

 >-------------+-------------------+---+------------------+------------------><
               |                   |   |                  |
               +-| SELECT Clause |-+   +- | SORT Clause |-+
```

Filter_Expr:

```
 >--------+-----+-- list_col ---+-------+-- <op> --- value --+-----+--------->
         |     |               |       |                          |     |
         +- ( -+               +-- ~ --+                          +- ) -+
                               +-- ¬ --+
                               +-- \ --+
                               +- NOT -+
```

Description:

Specified as a CLI command or as a parameter on SELECT or SORT, the WHERE clause is used to restrict the rows displayed in the list.

A WHERE clause is a logical combination of one or more simple filter expressions that define search criteria used to select the required list rows. When applied to an individual list row, each filter expression in the where clause resolves to either 1 (true) or 0 (false). If the logical result of all the filter expressions is 1, then the row satisfies the where clause criteria. Only those rows which satisfy the where clause are displayed in the view of the list.

Use of a WHERE clause is not cumulative and so will replace any prevailing filter of rows in the display. It also resets any prevailing SORT clause specification. Columns displayed by the prevailing SELECT clause, are unchanged.

Executing a WHERE clause will add a new entry, which includes the prevailing SELECT clause, to the View List Display drop down menu.

Parameters:

`(    )    AND    OR    NOT`

Logical operators supported by the where clause and supported symbolic equivalents are as follow:

( )	Left and Right Brackets (parentheses - X'4D' and X'5D').
AND	& (ampersand - X'50')
OR	¦ (broken bar - X'6A'), | (vertical line - X'4F')
NOT	~ (tilde - X'BC'), ¬ (not sign - X'5F'), \ (backslash - X'E0')

Notes:

1. To avoid confusion, it is recomended that parentheses be used where more than two filter expressions are specified in order to establish logical AND/OR precedence.
2. Parentheses must be balanced so that there are an equal number of left and right parentheses in the where clause.

Filter_Expr

A filter expression that tests a single column within the list display.

The contents of *list_col* are tested against a test value specified by *value* to establish a TRUE (1) or FALSE (0) condition.

list_col

The name of list column whose contents will be tested in this filter expression.

Where *list_col* is a numeric field, *value* must be numeric and an arithmetic compare is performed. Similarly, if *list_col* is a character field, then *value* is treated as a character string with blank padding on the right. Check the field Type in the FDB to determine whether the field contains numeric or character data.

If the data type of the test value is not compatible with *list_col* then an error will be returned.

`~ | ¬ | \ | NOT`

The symbols ~ (tilde), ¬ (not sign) and \ (backslash) represent the logical NOT operator and reverses the TRUE or FALSE condition established by the comparison operator <op>.

<op>

Comparison operator specified as one of the following:

=	Equals.
<>	Not Equals.
<	Less than.
>	Greater than.
<=	Less than or equals.
>=	Greater than or equals.
<<	Contains. This operator applies to character columns only and returns TRUE if *value* is a sub-string of *list_col*.
>>	Begins. This operator applies to character columns only and returns TRUE if *value* is a sub-string at the start of *list_col*.

value

The value to be tested against the contents of *list_col*.

If *list_col* is a character field, *value* is treated as a case insensitive character string unless it is enclosed in (') apostrophes or (") quotation marks. Apostrophes or quotation marks ensure case sensitivity and are mandatory if *value* is the same as a list column heading or if it contains any of the comparison operators or blanks.

SELECT Clause

Any valid SELECT clause used to select column headers.

SORT Clause

Any valid SORT clause used to sort list rows.

Example:

```
where    col1 = 'A'   and   (col2 << 'C' or col3 > 4)
```

List only those rows for which col1 is equal to A and either col2 contains C as a substring or col3 is greater than 4.

Note: the quotes around the literal strings are not needed unless there are columns in the list with name A or C and that the blanks separating the elements of the expression are optional.

The Sort (Order By) Clause

Syntax:

```
                      +---------- , -------------+
                      |                 +- A -+  |
                      v                 |     |  |
>>--+-- SORT -----+---+-- column_name --+-----+--+-------------------------->
    |             |                     |     |
    +-- ORDER BY -+                     +- D -+

 >-------------+-------------------+---+-------------------+----------------><
               |                   |   |                   |
               +-| SELECT Clause |-+   +- | WHERE Clause |-+
```

Description:

See also Sorting with the Cursor.

Specified as a CLI command or as a parameter on SELECT or WHERE, a SORT clause is used to modify the order in which the rows are displayed in the list.

Use of a SORT clause is not cumulative and is based on the list's default sort order. Therefore any prevailing sort order is ignored. The prevailing SELECT and WHERE clauses are unchanged.

Unlike SELECT and WHERE, a SORT clause specified as a CLI command is not added to the View List Display drop down menu.

Parameters:

column_name

Name of the List Window column on which to sort.

Column names must be supplied in the order in which they appear in the list heading (or in the FDB).

A | D

The sort order is specified with a list of column sort specifications which consist of a column name followed by a sort direction. The sort direction is given as A for ascending or D for descending.

If the sort direction is not provided, it defaults to ascending.

Commas must be used to separate multiple column sort specifications whereupon intervening blanks are permitted.

`SELECT Clause`

Any valid SELECT clause used to select column headers.

`WHERE Clause`

Any valid WHERE clause used to filter list rows.

Sorting with the Cursor

A quick way of sorting a list view is to place the cursor on the heading of the column on which you want to sort and then press the <Enter> key. Alternatively, if configured, simply double-click the left mouse button on the list column header.

For fields that contain timestamps and/or datestamps, the first time the sort is actioned on the column header, the data is arranged in descending order. For all other column fields, the data is initially arranged in ascending order.

Subsequent sorting on a column header using this method will reverse the order in which data in that column was last sorted.

List Entry Location

If the first field column of the list meets the requirements detailed below, then the occurrence of this field in the first row of the display area (list current row) is marked as being enterable (text is yellow underscore). Simply overtyping this field with the required field string value will scroll the display so that the first row containing this value becomes the new current row.

1. The column is keyed. (Key=Yes)
2. The column contains character data. (Type=Char)
3. The column's fields are sorted in ascending order.

This information is available from the FDB.
Unless a SELECT clause has been executed which changes the first column of the display, these criteria are usually always met by list windows of class LISTFILE.

Partial strings may be entered but residue from the overtyped field should be deleted or blanked over, otherwise it is included as part of the typed string. Similarly, strings entered in lower case are automatically upper cased.

Starting at the row following the list current row, field values occupying the first column are each compared with the string value entered by the user until one is found which is greater than or equal to this string value.

The target row is the row containing an exact match for the specified field string value or else the row immediately before a row containing a field value which is greater than the specified value. The list is scrolled so that the target row becomes the list current row.

Location of list entries is also achieved using the following:

- FIND/RFIND Command
- LOCATE Command
- S Command

FIND Command

Syntax:

```
>>-- Find -------- string ---------------------------------------------------><

>>-- RFIND ------------------------------------------------------------------><
```

Description:

Find is used to scroll the display to the **next** list entry to contain the specified search string **anywhere** within a field in the the first column of the display. If no match is found for **string** then no scrolling occurs.

Following a successful FIND operation, `RFIND` (assigned to PF5 by default) may be used to repeat the search for the remaining list entries.

FIND *string* is only valid if the first column in the display is defined as being a character key field. (Check the FDB for Type=Char and Key=Yes).

Note that key fields area highlighted and remain at a fixed position within the display when scrolling left and right. If multiple key field columns exist within the list, then changing the order of the key fields using a SELECT clause, will allow the user to execute

FIND/RFIND on the contents of an alternate key field column.

Parameters:

string
: A character search string used in the compare against data within fields in the first column of the display.

LOCATE Command

Syntax:

```
>>-- Locate ------ string ---------------------------------------------------><
```

Description:

Starting at the first entry and proceding downwards, LOCATE will compare *string* against data at the **start** of each field in the the first column of the display until either a match is found or the field data is greater than *string*.

If the strings are equal, then the display is scrolled so that this list entry becomes the first row in the display. Otherwise, if the list entry field data is greater than *string*, the display is scrolled so that the first row in the display is that which immediately precedes this list entry.

`LOCATE` *`string`* is appropriate only if the first list column is in ascending sort order and is only valid if the first column in the display is defined as being a character key field. (Check the FDB for Type=Char and Key=Yes).

Parameters:

string
: A character search string used in the compare against data at the start of fields in the first column of the display.

S Command

Syntax:

```
>>-- S ----------- member ---------------------------------------------------><
```

Description:

Supported as a CLI line command for Library Lists only, S *member* will perform the default operation (i.e. Edit) on the specified library member.

S is also supported as a List window prefix command which applies to all types of List window. In this case, S will execute the default operation for the particular list entry type. See the relevant prefix command table <Dflt> entry for for each of the supported Execute CBLVCAT, List and File Search windows.

Parameters:

member
: The library member name.

List Window Prefix Area

List windows provide support for a prefix area which is displayed as a one to eight character enterable field occupying the first column of a list. This field is displayed with underscore characters to indicate that it is enterable.

A command entered in the prefix area of a list entry is actioned when a PFKey or <Enter> is hit. i.e. On a single 3270 I/O. (If configured, double-clicking the left mouse button will also action the command.)

Only one prefix command may be actioned against a list entry in a single 3270 I/O. However, multiple prefix commands may be actioned for multiple list entries in a single 3270 I/O. Where prefix commands are entered against multiple list entries, each command is executed in order from the top of the list to the bottom.

Supported prefix area commands depend on the individual list window application. A decription of supported prefix commands is documented in the help for each list window application.

A summary of all list prefix commands supported by parent windows of window class LISTFILE and VCIWEXEC, is found in Appendix B - List File Prefix Command Summary.

Default Action

Whether or not a list has a prefix area, list windows usually have a default action performed when a list entry is selected as if an item in a menu. i.e. When the cursor is positioned on a list entry and <Enter> is hit or, if configured, the left mouse button is double-clicked on the list entry. This is equivalent to executing prefix command "S" against a single list entry.

Help documentation for a list window applications that supports this default action includes a description of the default action as the <Dflt> entry in the application's table of supported prefix commands. e.g. CBLVCAT Interactive, List Dataset and File Search window prefix command tables.

Note that the default action is disabled if a prefix command is entered against any entry in the list.

Unrecognised Prefix Commands

Entering an unrecognised prefix command will return error ZZSW002E.

However, for file lists, where prefix command "E" (Edit) is supported, command, edit macro or procedure names may be entered in the prefix area. In this case, where the command entered in the prefix area is not one of the supported prefix commands, it is passed back through the window hierarchy in the following order. The command is processed by the first window or environment in which the command is recognised.

1. The CBLe text editor, if the list window is a CBLe List document window. (i.e. a CBLe synonym, command or macro name.)
2. The SELCOPY/i environment command processor. (i.e. a SELCOPY/i command.)
3. TSO, if SELCOPY/i is started in ISPF or native TSO. (i.e. a TSO command, CLIST or EXEC.)

The full fileid of the entry against which the command is executed, is passed as the only input parameter to the command.

Repeating Prefix Commands

The last prefix command executed in the current list window may be repeated for any list entry by entering "=" in the prefix command area.

If other list entry prefix commands are executed in the same 3270 I/O before attempting to execute the command for this entry, then the command executed will be the last command executed for an entry in the list that occurs before this entry. Otherwise, the prefix command to be repeated is the last one executed on a previous 3270 I/O for the current list window.

Executing Prefix Command on a Block of List Entries

Any command entered in the prefix area may be repeated for each entry in a block of list entries marked by "//" in the prefix area of the first and last entries of the block. The command (or macro, etc.) to be executed must be specified on either the first or last entry of the block, following the "//" marker. e.g. "//d" in the prefix area of a file entry and "//" in any subsequent (or previous) file entry's prefix area defines the start and end of a group of file list entries that will be deleted.

Multiple, non-overlapping blocks of list entries may be marked, each block executing a specified command. If an unbalanced pair of "//" block markers is found or no command is specified on a block of list entries, execution of all blocked entry commands is delayed until there are and even number of block markers, each block having an accompanying command.

Interactive Panel Windows

Interactive panel (IPO) windows are of window class, WINWIPO0, and are used primarily for option, dialog and list windows.

SELCOPY/i functions that utilise interactive panel windows include:

- DB2
- File Copy
- File Compare
- File Search/Update/Copy/Remap
- Favorite Datasets/Commands
- SDE Edit

In addition to standard window features, windows of window class WINWIPO0 support a number of unique features that are described in this section.

- Panel Window Format
- Panel Window Size & Location
- Panel Window Hierarchy
- Panel Scrollable Display
- Panel Window Views
- Scrollable Input/Output Fields
- Input Field Data Recall
- Embedded Tables

Panel Window Format

In addition to the Command> and Scroll> prompts, an IPO window view may contain any of the following panel elements.

Panel ID
Every IPO window has a unique panel id which is displayed in a non-scrollable line of data at the top of the window display, below the window menu bar (if present).

Menu Bar
A menu bar may be displayed at the top of the panel display area below the window title bar. If the panel view includes an embedded list, then these menu items will be as described by the list window menu for the List window class.

Lines *n_first-n_last* of *n_total*
If a panel view does not include an embedded table or list, this information is displayed, right-adjusted, on the same non-scrollable line as the panel id. It identifies the line numbers of the first (*n_first*) and last (*n_last*) lines currently on display, and also the total number of lines (*n_total*) in the panel view.
e.g. "Lines 1-12 of 24"

This information is displayed because, by default, IPO panel windows are scrollable. The exception occurs where panel views contain an embedded table or list in which case the table/list entries are scrollable, not the panel display.

Input Fields
Input fields are enterable fields which allow the user to configure the functions performed by the panel. A value entered in a field must conform to that field's defined data type (e.g. numeric or text).

Where valid input field text can exceed the width of the input field area within the panel display, the input field may be scrolled and expanded as required. (See Scrollable Input/Output Fields)

Text input fields may have been defined so that the specified input value can be only one of a pre-determined list of values. If the text entered into an input field of this type does not exactly match one of these values, then the first value in the list that begins with the inputted text will be selected. If no match is found, then the input text is considered to be invalid and so a pop-up window is opened from which the user can select one of the valid input values from a list of possible values.

In a single panel, one or more related input fields may have been defined to dependent upon, and have values identified by, column values in a single row of a list that is not pre-determined by the program. i.e. the list of possible input field values is generated at run-time and displayed in a modal panel window containing an embedded list.

Any change to one of the dependent input field values will invoke this panel so that list entries are generated from some other source (e.g. a DB2 catalog table) and filtered by value(s) that have been entered in these input field(s). Simply selecting a row from the list panel will populate the dependent fields in the parent panel with relevant values from the list row and afterwards close the list panel.

The wild card character '%' (percent) or '*' asterisk, each representing zero or more characters, may be included in the dependent input field value in order to filter the list of possible values. If a value is entered (with or without the wild card character) that identifies a unique row in the generated list, then display of the list window is bypassed and the input field(s) populated accordingly.

When a panel is opened, input fields may contain values that have been determined using the following search sequence:

1. A field input value as specified by the user via a panel invocation CLI command. The DB2 CLI command opens a DB2 panel hierarchy and optionally populates panel input fields. e.g. `DB2    7.4    CREATOR=ZZS`

2. A default field input value as specified in the panel definition source. A default field input value is often defined to match default values in underlying syntax. e.g. DB2 default precision value for a DECIMAL field is 5.

3. If the field is configured to do so, the value displayed is that entered in the input field the last time the panel was opened. The value of each panel field is saved as a SELCOPY/i INI variable when focus is removed from the panel window, and in the SELCOPY/i User INI file when the SELCOPY/i session is ended normally.

Output Fields
Output fields display variable data that is not part of the panel's static text and is included in the panel for information purposes only. These types of field are non-enterable and usually contain values that are generated as a result of previously supplied user options or input field entries.

Like input fields, output fields may be scrollable and expanded.

Option Check Boxes
An option check box is a single character input field which is either set on (checked) or off (unchecked). Any non-blank character may be entered in a check box to select that item, however, when the display is refreshed, the check box character will displauy as "/" (slash).

Comment text that accompanies a check box, describes the action taken when the box is checked. Further information may also be found in the panel help.

Radio Buttons
A radio button group is a collection of 2 or more mutually exclusive check boxes, one of which is always checked. When a radio button is checked, all other buttons in the radio button group become unchecked.

In the event that the user checks more than one radio button in the group before hitting <Enter> (or a PFKey), then the newly checked radio button that is positioned closest to the bottom right hand corner of the 3270 display, will be selected.

Options Menus

An option menu defines a list of numbered items from which the user may select a single item.

If other input fields exist in the panel view, an option menu usually has an accompanying option menu entry field in which the user enters a menu selection.

An item is selected using either of the following methods:

◊ Position the cursor on the required item and select <Enter> or, if configured, double-click the left mouse button on the item.

◊ If present, enter the menu item number in the option menu entry field provided, otherwise, enter the item number at the command line prompt.

Embedded Tables

IPO windows may include independently scrollable, embedded tables that have similar characteristics to the display of structured data in an SDE edit view. e.g. Prefix commands and PFKeys assignments that zoom, insert, delete and replicate table entries.

Embedded tables provide a method of supporting multiple, repeating groups of input/output fields. Unlike embedded lists, embedded tables may contain editable fields.

Embedded Lists

An IPO window may contain embedded lists based upon input parameters entered by the user. The list entries are not generated until the user has completed input of, or accepted existing values for the parameter fields and hit <Enter>.

IPO windows that include embedded lists adopt characteristics (menu items, commands, etc.) as provided by list windows.

Panel Window Size & Location

Panel windows may be resized and repositioned using standard SELCOPY/i window manipulation techniques. The last customised position and size of an IPO panel window is stored in the SELCOPY/i User INI file and is restored to this position and size the next time the panel is opened.

The stored window size and position persists across subsequent SELCOPY/i sessions.

Panel Window Hierarchy

An IPO panel window may be opened as a dependent window which is owned by another IPO panel. Furthermore, the process may be recursive so that an owned IPO panel window may itself be the owner of another IPO panel window. In the case of an expanded input field, the resulting text edit window view is owned by the IPO panel containing the input field.

Although it may still be in view, window focus cannot be placed on an owner IPO panel until the panel it owns is closed.

In addition to being owned, IPO panel windows containing embedded lists for selection of input field values are also modal.

Panel Scrollable Display

Unlike standard dialog windows, panel windows support up/down scrolling of data displayed in the window. Left/right scrolling is supported for scrollable Input/Output Fields, embedded lists and embedded tables only.

Up/down scrolling is particularly necessary where a window display area is not sufficiently large enough to display the entire contents of the panel (e.g. following a window resize or for 3270 terminals with a low number of displayed rows.)

For panel views that do not include an embedded table or list, the lines of the panel currently on display in the window display area, is reported in a non-scrollable line at the top of the panel. e.g. "Lines 11-36 of 36"

The "Scroll>" input field is used to control scrolling type as described by commands UP and DOWN. Scroll UP and DOWN are assigned to <PF7> and <PF8> respectively.

Unless positioned Scrolling performed on panel views that include an embedded table or list, will scroll the contents

Panel Window Views

A function associated with an IPO panel window may require a large amount of user input. If so, panel text and user input may either be displayed in a single, scrollable panel view, or split over a number of separate views within the same panel window, where each view contains related information.

Each panel window view may be considered an extension to user options and input fields presented in the first (primary) panel view.

Secondary panel window views are often displayed as a result of selections made in the primary panel view. The VIEW command may also be used to scroll between panel views. By default, "VIEW +" and "VIEW -" to scroll forwards and backwards through the panel views are assigned to <PF19> and <PF20> respectively.

Closing a secondary panel view using command, END (assigned to <PF3> by default), will return panel focus back to the primary panel view. If END is executed at the primary panel view, then the IPO panel window is closed.

Scrollable Input/Output Fields

Scrollable, expandable text input/output fields are supported allowing for input of more text than can be displayed in the field area.

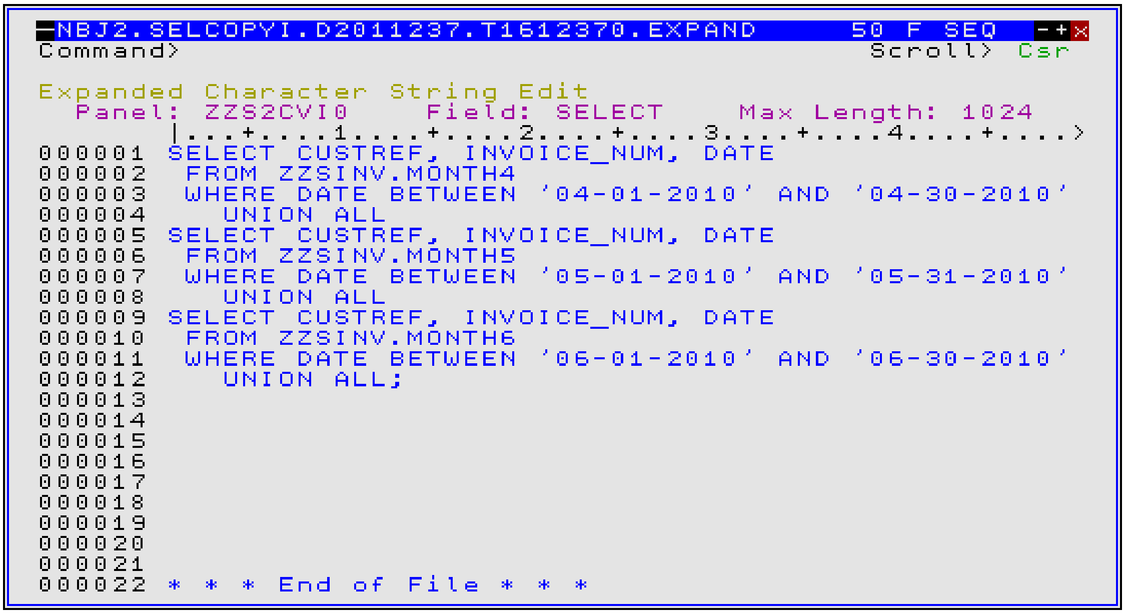
These type of input/output fields are suffixed with a plus (+) and/or minus (-) symbol. The display of these symbols also provide an indication as to whether the field text in view is at the start ("+" only) or end ("-" only) of the field, or somewhere in between ("-/+").

To scroll the contents of the field, position the cursor within the field data and hit <PF10> to scroll left and <PF11> to scroll right.

The entire contents of the field may be expanded into a CBLe text edit view and so edited using the full functionality of the text editor, before being placed back in the input field when the text edit view is closed. To do this, position the cursor in the scrollable field and hit <PF2> which, by default, is assigned to command EXPAND.

The expanded field text edit view displays 50 character portions of the text string on each line of the edit area (i.e. text wraps at column 50 to column 1 of the next line.) For input fields, any text entered beyond column 50 will be ignored.

The panel and field name associated with the input/output field and the maximum length of the text string is displayed before the edit display area. Text entered beyond the maximum length of the string will be truncated.



```
=NBJ2.SELCOPYI.D2011237.T1612370.EXPAND      50 F SEQ  -+x
Command>                                         Scroll> Csr

Expanded Character String Edit
  Panel: ZZS2CVI0    Field: SELECT    Max Length: 1024
      |...+....1....+....2....+....3....+....4....+....>
000001 SELECT CUSTREF, INVOICE_NUM, DATE
000002  FROM ZZSINV.MONTH4
000003  WHERE DATE BETWEEN '04-01-2010' AND '04-30-2010'
000004    UNION ALL
000005 SELECT CUSTREF, INVOICE_NUM, DATE
000006  FROM ZZSINV.MONTH5
000007  WHERE DATE BETWEEN '05-01-2010' AND '05-31-2010'
000008    UNION ALL
000009 SELECT CUSTREF, INVOICE_NUM, DATE
000010  FROM ZZSINV.MONTH6
000011  WHERE DATE BETWEEN '06-01-2010' AND '06-30-2010'
000012    UNION ALL;
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022 * * * End of File * * *
```

Figure 8. Expanded input field text edit view.

Input Field Data Recall

The last entry entered in a panel input field is stored in the user's SELCOPY/i INI file when the panel is closed.

The next time that panel is opened, the contents of any input field may be restored to its last saved value by positioning the cursor within the input field and hitting <PF4> which, by default, is assigned to command, REMIND.

Embedded Tables

Overview

IPO window embedded tables provide a method by which multiple, repeating groups of input fields may be inserted, updated, replicated or deleted. An embedded table may also contain non-editable (output) fields which are included in the table display for information only.

The contents of a table may be populated from other sources governed by panel input fields (e.g. SQL queries, SELCOPY/i standard lists or other embedded tables.) Where a table contains rows of values to be used in generating the panel's function parameters, then populated table rows that are not required may be deselected simply by excluding them from the display.

Entries in a column of an embedded table may each represent rows of another embedded table. If so, each column field entry is coloured, by default, in yellow underscore with a ">" (greater than) prefix with a number of embedded table rows specification (e.g. `> 2 specified`.) Placing the cursor on a field entry of this type and pressing <Enter> will display an IPO sub-panel containing the embedded table represented by the parent table column field. On ending the sub-panel (<PF3>) the parent column entry will be updated to reflect the number of table rows specified in its child table.

Users should be aware that the order in which the table rows occur are the order in which parameters are generated for the panel's underlying command syntax. Although always syntactically correct, performance benefits may exist if parameters generated for table rows are specified in a particular order. e.g. Record key fields provided for Compare Files key synchronisation.

Table row entries may be edited using a set of command line (primary) commands, prefix (line) commands and pre-defined PFKey assignments that are analagous with structured data edit (SDE) commands and PFKey assignments.

Table Edit CLI (Primary) Commands

The following table contains command line interface (CLI) commands that may be executed by the user to edit tables embedded in IPO panels. The syntax specification for each command is as documented for the SDE command equivalents.

ALL	Synonym for WHERE.
BOTTOM	Display the last page of table rows. (Equivalent to DOWN MAX)
DELETE	Delete table rows. (Default is focus row)
DOWN	Scroll down through the display of table rows.
FLIP	Flip disply of excluded and non-excluded table rows.
INSERT	Insert table rows.
LEFT	Scroll left through the display of table columns.
LESS	Filter (exclude) additional table rows.
MORE	Filter (include) additional table rows.
QUERY	Query current table edit options.
REDO	Redo changes to the table that have been undone.
RIGHT	Scroll right through the display of table columns.
SELECT	Select table columns and their order of display.
SET	Set table edit options.
TOP	Display the first page of table rows. (Equivalent to UP MAX)
UNDO	Undo changes made to the table.
UP	Scroll up through the display of table rows.
WHERE	Filter table rows.
ZOOM	Zoom a table row to display it in single record view.

Table Edit Options

The following table contains available table edit SET and QUERY options.

COLHEADER	Display descriptive or internal format column header names. (See use of COLHEADER in Table Editing Techniques below).
COLOUR, COLOR	Colour specification for individual areas of the display.
COLWIDTH	Display width assignment for individual table columns.
MSGMODE	Controls display of messages.
MULTIPOINT	Controls support of >1 line label name assignment for any row.
PFKEY	PFKey assignment.

POINT	Line label name assignment.
PREFIX	Prefix area display settings.
REFERENCE	Table column field reference header line display.
SCALE	Table column scale header line display.
SHADOW	Display of shadow lines representing a group of excluded rows.

Table Edit Prefix (Line) Commands

.name	Set a line pointer (line label name).
A	Make this row the target for a move or copy (move or copy rows After this row).
B	Make this row the target for a move or copy (move or copy rows Before this row).
C(n) CC	Mark a row or a block of rows for copying. Rows may be copied to another position within the same table using prefix commands, A or B.
D(n) DD	Delete a row or a block of rows.
F(n) F*	Include the first *n* rows of an excluded row group. F* include all excluded rows.
I(n)	Insert a new row or a block of *n* new rows.
L(n) L*	Include the last *n* rows of an excluded row group. L* include all excluded rows.
M(n) MM	Mark a line or a block of lines for move. Lines may be moved to another position within the same table using prefix commands, A or B.

R(n)
RR(n)
"(n) ""(n)
	Replicate (duplicate) a row or a block of rows *n* times.

X(n) X* XX	Exclude a row or a block of rows from the display. X* exclude all rows from the current row to the last row of the table.
Z	Switch to a zoomed (single record view) display of the row.

Table Edit PFKeys

The following PFKeys are assigned by default when the cursor is within the embedded table display area.

PF01	INSERT. (Inserts a new table row)
PF02	ZOOM. (Display a row in single view format.)
PF07	UP. (Scroll up.)
PF08	DOWN. (Scroll down)
PF09	MDINEXT. (Pass focus to the next window.)
PF10	LEFT. (Scroll left)
PF11	RIGHT. (Scroll right)
PF13	DELETE. (Delete a row.)
PF14	DUPLICATE. (Duplicate/Replicate a row.)
PF21	ISPF SWAP LIST. (Display the ISPF swap list menu.)
PF22	UNDO. (Undo a level of table edit changes.)
PF23	REDO. (Redo a level of table edit changes undone by the last UNDO operation.)

Table Editing Techniques

The following hints and tips relate to tasks commonly performed on IPO panel embedded table fields

Single Row Display

Default display of an IPO embedded table is multi-row (table) format where **UP** (<PF7>) and **DOWN** (<PF8>) scrolls the table rows and **LEFT** (<PF10>) and **RIGHT** (<PF11>) scrolls the table columns.

All column names and values belonging to a single table row, may be displayed in the panle view simply by placing the cursor on a table row and pressing <PF2> or entering prefix command "Z" to execute **ZOOM** and so display that row in single row format. <PF2> again will return to the table display format. Column field values that are ediatble in multi-row format may also be edited in single row format.

Whilst in single row display format, use <PF10> and <PF11> to scroll backwards and forwards (i.e. LEFT and RIGHT) respectively through the table rows.

In some instances, the panel table will have been defined so that the zoomed display of the table fields is a formatted entry form containing explanitory text. This zoomed view of the table row is displayed in another panel view, in which case the END command (<PF3>) should be executed in order to return to the table view.

Filtering the Table Rows

A number of rows (n) may be manually excluded from display using the prefix (line) command "Xn" or, to mark a block of rows for exclusion, "XX".

WHERE, **MORE**, and **LESS** CLI (primary) commands may also be used to include/exclude multiple table row entries mechanically, based on the contents of any of the displayed fields. (**ALL** is a synonym for WHERE.)

These commands operate on structured data (SDE) expressions which support operators, character strings, numeric values, built-in functions, sub-expressions and *field values*.

Embedded table field values may be referenced in an expression via the column (field) reference number (e.g. #1, #4) or via the internally defined column (field) name.

By default, the column reference numbers and internally defined column names are not displayed. This is to maximise the amount of table data in view and to display more meaningful column headers provided by the column titles.

To display and then remove from display the field reference numbers, use command **SET REFERENCE ON**/**OFF** (abbreviated to **REF ON**/**OFF**). Similarly, to alternate the column header display between the column title an its internally defined name, use command **SET COLHEADER NAMES**/**TITLES** (abbreviation COLH N/T).

Examples:

```
LESS SelectTyp='AN'
```

Additionally exclude all entries where the value of table field name "SelectTyp" is "AN". Entries that were already excluded will be unaffected.

```
WHERE SelectFld >> 'ABC-' or #1 << 'DEF-'
```

Exclude all entries except those where the value of table field name "SelectFld" begins with literal "ABC-", or the value of field reference number 1 contains literal "DEF-".

Re-Ordering the Table Columns

By default, table columns are presented to the user in a logical order.

However, the user may suppress and/or change the order of columns displayed using the command **SELECT**, which specifies table columns by column (field) reference number or via the internally defined column (field) name. (See filtering techniques above for display of these column field attributes.

To redisplay columns in their default order, enter "SELECT *".

Adding/Inserting Table Rows

Table rows may be inserted using the INSERT (primary) command, "I(n)" prefix (line) command or by positioning the cursor on th erow befored the inserted row and pressing <PF1>.

Table rows may be replicated using the "R(n)" or "RR" prefix (line) commands or copied using prefix (line) commands "C(n)" or "CC" combined with "A" (after) or "B" (before).

The order in which table rows occur may be important. To re-order the table rows, use prefix (line) commands "M(n)" or "MM" combined with "A" (after) or "B" (before).

Undo/Redo Changes

Edited changes to tables (including field value updates, row inserts, deletions, etc.) may be undone and, if required, re-applied using the commands **UNDO** and **REDO** respectively.

Alternatively, the cursor may be positioned within the table area and then <PF22> or <PF23> pressed to perform UNDO or REDO respectively.

Undo levels are maintained for the table even if the panel view is changed and re-visited. However, on closing the panel the table rows are dropped from storage so that restarting the panel will reinitialise the table rows within the IPO panel view.

COLHEADER - SET/QUERY Option

Syntax:

```
>>-+-----------+-- COLHeader -------+-- Name ---+----------------------------><
   |           |                    |           |
   +- SET -----+                    +-- Title --+

>>--- Query ------ COLHeader ------------------------------------------------><
```

Description:

This option controls the display format of the table column names header line.

By default, column are displayed with their descriptive column titles. However, when referencing columns on a SELECT command or in an expression for WHERE, MORE or LESS row filtering operations, then the column's internal format name or its column field reference id must be used. For this purpose, SET COLH NAME may be used to display the columns with their internal name format.

SET Value:

`NAME | TITLE`

Display all table columns with their internal column name format (NAME) or with their descriptive column title (TITLE).

QUERY Response:

The column header names format (NAME or TITLE).

Primary Option Menu (=)

The Primary Option Menu panel (ZZSGPRIM) is an interactive panel window providing an entry point to all SELCOPY/i panels and functions.

This panel may be started by enetering "=" (equals) at any command line. A fast path may be specified immediately following the "=" symbol to directly open sub-panels of the Primary Option menu. (e.g. =0.4.1 for "COBOL Compiler options.")

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel.

`SwapList`

If SELCOPY/i is operating within an ISPF split screen, opens the ISPF task list of active ISPF logical sessions.

`Window`

Opens the Window List window containing a selectable list of all open windows in the SELCOPY/i session.

`Help`

Open the general help for the Primary Option menu panel.

Options

0 Settings	Set SELCOPY/i options
1 Text Edit	Edit/View small text-type files
2 Data Edit	Edit/Browse potentially large data files
3 List	List Volumes,VTOCs,Datasets,Members etc
4 **Home**	Edit and execute point-and-shoot commands
5 Copy/Reformat	File Copy with optional copybook reformat
6 Search/Update	File Search/Update/Copy/Reformat
7 Compare	File/Library Compare Utilities
8 Utilities	General utilities
9 Structure	Create structure from copybooks etc
10 Filter	Create record selection filter
11 Alloc/Define	Create new VSAM or Sequential datasets
12 DB2	Work with DB2, browse/edit tables etc
W Window List	Display active windows, select with cursor to switch focus
X **Exit**	Exit SELCOPY/i

Panel Output Fields

`User:`

An output field displaying the user's logon id.

`Version:`

An output field displaying the version of SELCOPY/i.

`Date:`

An output field displaying the current date.

`Time:`

An output field displaying the current time.

`OpSys:`

An output field displaying the operating system release.

`System:`

An output field displaying the z/OS system name as defined by the SYSDEF statement in active parmlib member IEASYMxx.

`VM User:`

For VM guest operating systems or CMS users, this output field displays the VM userid of the guest machine or CMS system.

Settings (=0)

The Settings panel (ZZSGSET0) is an interactive panel window, opened on selection of option 0. in the SELCOPY/i Primary option menu.

This panel, and its sub-panels, establish default options and values applicable to the individual user's SELCOPY/i environment. These values are assigned to variables set in the user's own SELCOPY/i User INI file.

Options

Individual option entries relate to specific functions of the SELCOPY/i environment.

1 Startup	Startup options
2 System	System options
3 Text Edit	Text Editor (CBLe) options
4 Data Edit	Structured Data Editor (SDE) options
5 List	List window options

Panel Input/Output Fields

`REXX Macro Library Path:`

Fields that together establish the library search chain used to locate a SELCOPY/i REXX macro.

`User Library>`

An input field allowing the user to enter the fully qualified name of one or more PDS/PDSE SELCOPY/i REXX macro libraries to appear first in the macro path search chain.

This field may be expanded (using <PF2<) in order to enter any number of blank delimited library data set names in the order in which they are to appear in the search chain. Close the expanded display (<PF3<) and press <Enter< to refresh the count of user libraries (#1 of n).

Note that, in any single update of the User Library field, if a non-existant library name is specified, then message ZZSE062E "Invalid macro path" is returned, the update is not applied and the original User Library field value is reset to its state prior to attempting the update.

`Site Library>`

A non-enterable (output) field identifying the DSN of the penultimate library in the macro search path.

This library contains macros that have been developed at the client's installation and made available to all SELCOPY/i users.

`CBL Supplied Library>`

A non-enterable (output) field identifying the DSN of the last library in the macro search path.

This library (SZZSDIST.CBLE) contains macros that have been distributed by CBL as part of the SELCOPY Product Suite and contains macros available to all SELCOPY/i users.

Startup Settings (=0.1)

The Startup Settings panel (ZZSGSET1) is an interactive panel window, opened on selection of option 1. in the SELCOPY/i Settings panel.

This panel specifies which application windows are to be opened at startup of SELCOPY/i.

Panel Input Fields

`Primary Option Menu`

Select this option field to open the SELCOPY/i Primary Option menu at startup.

`Home File`

Select this option field to open a CBLe text edit view of the user's HOME (CMX) command centre file.

System Settings (=0.2)

The System Settings panel (ZZSGSET2) is an interactive panel window, opened on selection of option 2. in the SELCOPY/i Settings panel.

This panel specifies options relating to SELCOPY/i general operation.

Panel Input Fields

`Command Line>`
Specifies the location of the command line in all SELCOPY/i windows to the TOP or BOTTOM of the display.

`Command Delim>`
Identifies the single character interpreted as the command separator used to enter multiple commands from a single command prompt.

`SDSF FastPath>`
Specifies the fast path that may be entered at the ISPF Primary menu panel in order to start SDSF (or an equivalent product) in an ISPF environment.

This option is used by SELCOPY/i OQ and OP macros to display JES2/JES3 job queues and the operator console respectively.

`Abend Trap>`
Specifies whether the SELCOPY/i abend trap is activated to trap any SELCOPY/i internal system abends and, if possible, recover from the abend.

If set ON, the abend will be trapped and a formatted dummp written to a SELCOPY/i dump data set. This dump data set may be requested by CBL for diagnostic purposes.

`Dump Prefix>`
Specifies the data set name prefix (maximum length 26) to be used for a generated SELCOPY/i formatted dump data set.

Qualifiers of the form '.Dyyyyddd.Thhmmssx', representing the current date and local time, are appended to the dump data set name prefix qualifiers.

Text Edit Settings (=0.3)

The Text Edit Settings panel (ZZSGSET3) is an interactive panel window, opened on selection of option 3. in the SELCOPY/i Settings panel.

This panel specifies options relating to the SELCOPY/i text editor.

Panel Input Fields

`Interface>`
Specifies the default edit interface.

The SELCOPY/i text editor supports edit commands supported by both the ISPF editor and the CMS XEDIT/Windows KEDIT editors. Some command verbs exist for both editors but can have very different effects (e.g. CHANGE). The prevailing SELCOPY/i text edit interface dictates the precedence by which common command verbs are interpreted and also influences the screen display and scrolling.

`Size Warning>`
Specifies the file size threshold at which the SELCOPY/i text editor will warn the user that it is about to load all records of a large file into storage. The message also prompts the user to either continue with the load or switch to using structured data edit which supports edit without all records being loaded in storage.

This value may be specified as a number of bytes (nnn), kilobytes (nnnK) or megabytes (nnnM).

`Load Warning>`
Specifies the file load warning threshold. During load of a file for text edit, when the number of bytes loaded reaches a factor of this load warning threshold, then a message is displayed prompting the user to continue or cancel the file load.

This value may be specified as a number of bytes (nnn), kilobytes (nnnK) or megabytes (nnnM).

Structured Data Edit (SDE) Settings (=0.4)

The Structured Data Edit (SDE) Settings panel (ZZSGSET4) is an interactive panel window, opened on selection of option 4. in the SELCOPY/i Settings panel.

This panel specifies options relating to the SELCOPY/i Structured Data (SDE) editor.

Options

1 COBOL	COBOL Compiler and Replacing Options
2 PL/1	PL/1 Compiler Options

Panel Input Fields

`Load Warning>`

Specifies the number of records loaded warning threshold. During load of a file for structured data edit, when the number of records loaded reaches a factor of this load warning threshold, then a message is displayed prompting the user to continue or stop the file load. If the load is stopped, then Update-in-place Edit is used.

This value may is specified as a number of records (nnn).

`Max Storage>`

Specifies the maximum storage available for SDE edit of a single data set.

An SDE edited data set is limited by the lesser of the prevailing MAXSTOR value and the amount of free private area storage above the 16MB line available within the region at the time of open. This limit is used to determine the SDE edit technique and data record management used to edit the data set.

This value may be specified as a number of bytes (nnn), number of kilobytes (nnnK) or a number of megabytes (nnnM). A value of 0 (zero) implies no maximum storage is to be applied.

`Aux Dsn HLQ>`

Specifies the dataset name prefix (maximum length 26 characters) to be used by SELCOPY/i SDE when allocating the default Auxiliary Edit data set. Auxiliary edit occurs when editing a non-KSDS data set that is larger than the Max Storage value or the calculated amount of free private area storage above the 16MB line.

Qualifiers of the form '.Dyyyyddd.Thhmmssx', representing the current date and local time, are appended to the dump data set name prefix qualifiers.

Default value is %USER%.CBLI.SDEAUX.

COBOL Compiler Options (=0.4.1)

The COBOL Compiler Options panel (ZZSGSETC) is an interactive panel window, opened on selection of option 1. in the Structured Data Edit (SDE) Settings panel.

This panel specifies options relating to the SELCOPY/i Structured Data (SDE) COBOL copybook support.

Panel Input Fields

`COBOL Compiler>`

Specifies the full DSN and member name of the COBOL Compiler module (e.g. IGY330.SIGYCOMP(IGYCRCTL).) SELCOPY/i will invoke the COBOL compiler when generating an internal SDE structure (SDO) from a COBOL Copybook.

Specification of a COBOL Compiler is necessary only if your COBOL compiler program module is not named IGYCRCTL and is not found in the library search chain.

`COBOL Max RC>`

Specifies the maximum acceptable COBOL compiler return code for which SELCOPY/i will continue to generate an SDE structure (SDO).

Where the COBOL compiler return code is greater than this value, the SDE create structure operation fails with an error message.

`COBOL Replacing Options:`

References 12 pairs of ('From:' and 'to:') fields that together generate a COBOL REPLACE statement which is inserted in the temporary source mamber used by SELCOPY/i as input to the COBOL compiler. This REPLACE statement is applied to **all** copy books selected for compilation.

The 'From:' field specifies a *pseudo-text* source string to be replaced. The corresponding 'to:' field specifies a *pseudo-text* replacement string.

PL/1 Compiler Options (=0.4.2)

The PL/1 Compiler Options panel (ZZSGSETP) is an interactive panel window, opened on selection of option 2. in the Structured Data Edit (SDE) Settings panel.

This panel specifies options relating to the SELCOPY/i Structured Data (SDE) PL/1 copybook support.

Panel Input Fields

`PL/1 Compiler>`

Specifies the full DSN and member name of the PL1 Compiler module (e.g. EL330.SIBMZCMP(IBMZPLI).) SELCOPY/i will invoke the PL1 compiler when generating an internal SDE structure (SDO) from a PL1 Copybook.

Specification of a PL1 Compiler is necessary only if your PL1 compiler program module is not named IBMZPLI and is not found in the library search chain.

`PL/1 Max RC>`

Specifies the maximum acceptable PL1 compiler return code for which SELCOPY/i will continue to generate an SDE structure (SDO).

Where the PL1 compiler return code is greater than this value, the SDE create structure operation fails with an error message.

List Window Settings (=0.5)

The List Window Settings panel (ZZSGSET5) is an interactive panel window, opened on selection of option 5. in the SELCOPY/i Settings panel.

This panel specifies options relating to the SELCOPY/i list windows.

Panel Input Fields

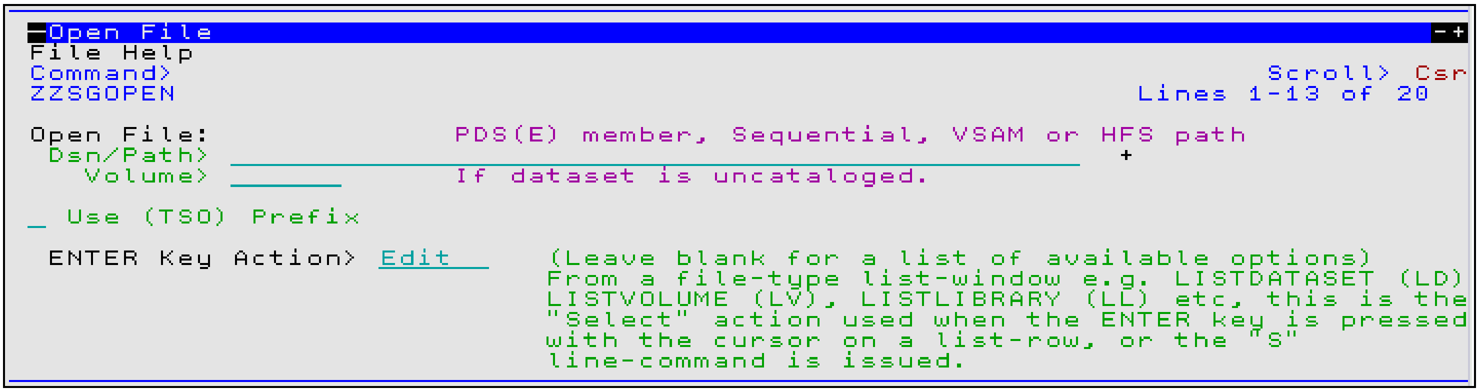
`ENTER Key Action>`

For file list windows only (window class LISTFILE), identifies the default action on pressing the <Enter> key on a list entry.

Possible actions are:

Edit	Open a CBLe text edit window to edit the file.
View	Open a CBLe text edit window to view the file (read-only).
Browse	Open a structure data edit (SDE) window to browse the file.
SDE	Open a structure data edit (SDE) window to perform full function edit of the file.
SDEU	Open a structure data edit (SDE) window to perform Update-in-place edit of the file.
None	Disable all actions on the <Enter> key.

Text Edit (=1)



```
 Open File                                                                  - +
File Help
Command>                                                            Scroll> Csr
ZZSGOPEN                                                        Lines 1-13 of 20

Open File:              PDS(E) member, Sequential, VSAM or HFS path
 Dsn/Path> ______________________________________________    +
   Volume> ______       If dataset is uncataloged.

_ Use (TSO) Prefix

 ENTER Key Action> Edit      (Leave blank for a list of available options)
                             From a file-type list-window e.g. LISTDATASET (LD)
                             LISTVOLUME (LV), LISTLIBRARY (LL) etc, this is the
                             "Select" action used when the ENTER key is pressed
                             with the cursor on a list-row, or the "S"
                             line-command is issued.
```

Figure 9. SELCOPY/i - Open File.

The Text Edit Open File panel (ZZSGOPEN) is an interactive panel window, opened on selection of option 1. in the SELCOPY/i Primary option menu.

This panel is used to open an existing cataloged or uncataloged data set PDS/PDSE library member or HFS file for CBLe text edit. For full documentation on text edit sessions, please refer to publication *"SELCOPY/i Text Editor (CBLe)"*.

Panel Input Fields

Open File:

Fields that identify the existing sequential or VSAM data set, HFS file or PDS/PDSE library member to be edited.

Dsn/Path>

An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library. Enclosing apostrophes (') may be used to ignore use of the TSO prefix if selected.

A selectable list of data set names or HFS files will be displayed as appropriate if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Volume>

Specifies a volume serial id mask for an uncataloged data set. (Not applicable to HFS files.)

Use (TSO) Prefix

Option field that controls whether the defined TSO prefix is used as the high level qualifier of the data set to be text edited.

Note that the TSO prefix will not be used if the data set name is enclosed in apostrophes (') or the fileid is an HFS file path.

ENTER Key Action>

Option field that controls the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button on an entry in the resulting file list.

Acceptable values are as follow:

Edit	Edit using the CBLe Text editor with Read/Write authority.
View	Edit using the CBLe Text editor with Read Only authority.
Browse	Browse data using the SDE Structured Data editor.
SDE	Full Edit of data using the SDE Structured Data editor.
SDEU	Update-in-place Edit of data using the SDE Structured Data editor.
None	No action to be taken. A list prefix command must be used instead.

Data Edit (=2)

Structured Data Browse/Edit Panel

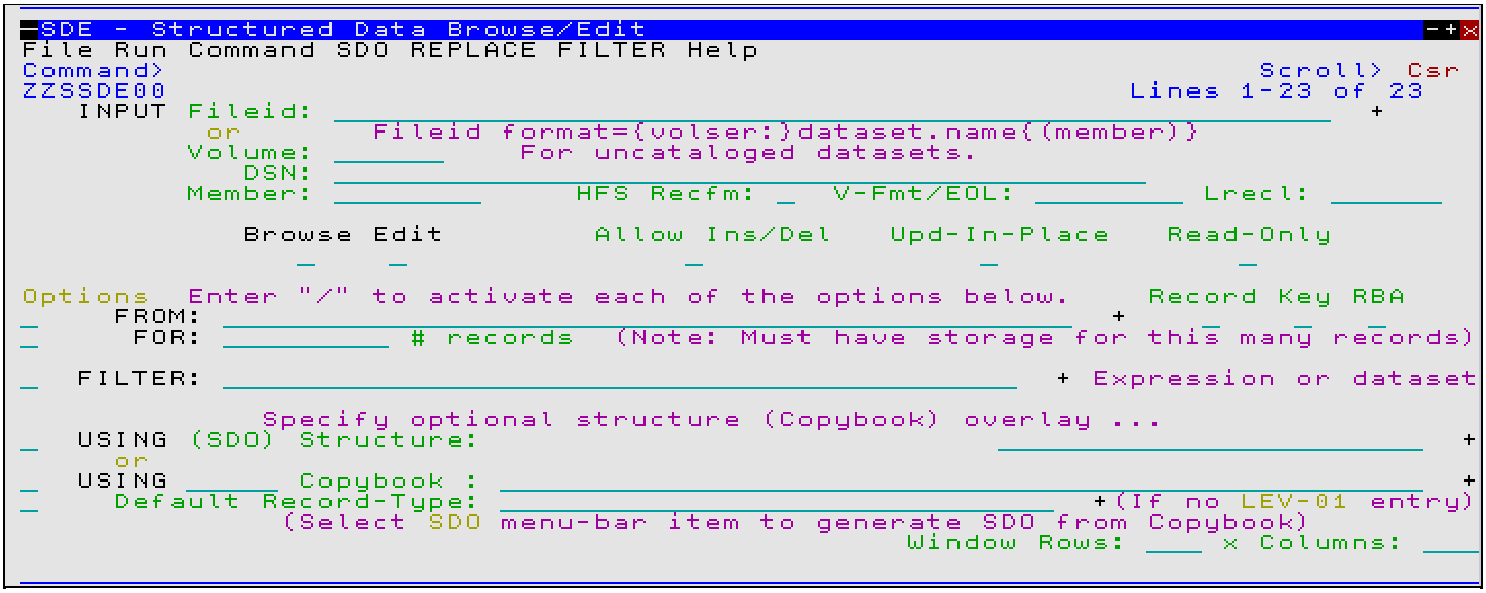
The Structured Data Browse/Edit Panel panel (ZZSSDE00) is an interactive panel window, used to open an existing file for Structured Data (SDE) edit.

For full documentation on the Structured Data Editor, please refer to publication *"SELCOPY/i Structured Data Editor (SDE)"*.

The panel may be opened via the following:

- Select option 2. 'Data Edit' in the SELCOPY/i Primary option menu.
- Select 'Structured edit' from the File menu in the main window menu bar.
- Execute the SDE command EDITDIALOG on the command line of an existing SDE window view, or from a SELCOPY/i text edit view if preceded by SDATA.
- Execute the command SDE with no parameters.
- Execute the prefix command "SD" from an Execute CBLVCAT or file list window.

The SELCOPY/i Structured Edit dialog window generates an SDE BROWSE or EDIT command to open an SDE window view in the current frame window.



```
 SDE - Structured Data Browse/Edit                                        -+x
File Run Command SDO REPLACE FILTER Help
Command>                                                           Scroll> Csr
ZZSSDE00                                                    Lines 1-23 of 23
   INPUT Fileid: _____________________________________________________  +
          or       Fileid format={volser:}dataset.name{(member)}
         Volume: ______       For uncataloged datasets.
            DSN: ___________________________________________
         Member: ________     HFS Recfm: _  V-Fmt/EOL: ________  Lrecl: ______

            Browse Edit         Allow Ins/Del    Upd-In-Place    Read-Only
               _    _                _               _              _

Options  Enter "/" to activate each of the options below.      Record Key RBA
_    FROM: ______________________________________________  +
_     FOR: _________  # records   (Note: Must have storage for this many records)

_  FILTER: ___________________________________________  + Expression or dataset

             Specify optional structure (Copybook) overlay ...
_  USING (SDO) Structure:                              ______________________   +
     or
_  USING _____  Copybook : ___________________________________________________  +
     Default Record-Type: ______________________________  +(If no LEV-01 entry)
              (Select SDO menu-bar item to generate SDO from Copybook)
                                                Window Rows: ___  x Columns: ___
```

Figure 10. Structured Data Edit Dialog Window.

Menu Bar Items

Run

Select this item or just press ENTER to begin immediate execution of the SDE BROWSE/EDIT command.

Command

Select this item to generate the SDE BROWSE/EDIT CLI command syntax corresponding to the entered parameters. The generated BROWSE/EDIT command is displayed in a temporary text edit window in a format suitable for execution using CMDTEXT (PF4).

The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

SDO

Select this item to generate the SDO (specified in the USING (SDO) Structure: field) from the copybook (specified in the USING Copybook: field).

Note that panel Create Structure from COBOL/PL1 Copybook(s) offers better flexibility in generating an SDE structure (SDO) from COBOL or PL1 copybooks.

REPLACE

The REPLACE menu item opens the COBOL Compiler Options panel to review and, if necessary, add COBOL REPLACE "From" and "To" pseudo-text values to be used in compiling a COBOL copybook.

Values enterd in this panel apply only to the current user. System wide COBOL REPLACE values may also have been entered in the SELCOPY/i Site INI file. (See the *"SELCOPY Product Suite Customisation Guide"* for details.)

FILTER

Opens the Create File Filter panel.

The dialog performs as a 'wizard' for generating a file-filter, and saving the selection criteria as a datset. The generated filter dataset name may subsequently entered into the "FILTER:" slot.

Help

Open the General Help for the SDE Structure Data Browse/Edit window.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time the Structured Edit dialog window was used.

Many field entries are optional and need to be activated by entering "/" in the preceding field. This provides easy deactivation/reactivation of a field value without having to clear the field.

INPUT Fileid:

Identifies the PDS/PDSE library mamber; sequential or VSAM data set, or HFS path to be browsed or edited.

Dataset names do not need to be enclosed in quotes, but must always be fully qualified.

Volume, dataset and PDS/PDS(E) member name may be specified as separate fields or altogether in this combination field e.g.

◊ USR1051.JCL(SELC022)
◊ USR1051.TEST.KSDS
◊ ZBIMS1:FRX310.FRX310T.SMPLOG
◊ OBS022:SYS2.LIVE.JCL(SELC022)

Volume:

Required for uncataloged datasets only, this field identifies the volume on which the dataset is saved.

DSN:

Fully qualified datasets name or HFS path name.

Member:

Member name of a PDS/PDSE library.

HFS Recfm:

For HFS files, optionally specify record format *F* or *V*.

◊ Specify *F* if the data is to be treated as fixed-length records, as defined by the **Lrecl:** field.
◊ Specify *V* if the data is to be treated as variable length records, as defined by the **V-Fmt/EOL:** field.

V-Fmt/EOL:

For variable length (RECFM=V) HFS files, optionally specify the variable format, or end-of-line character sequence used to determine the end of each record.

◊ *(offset,length,origin)*
RECFM V allows the user to specify the location of an embedded binary field that defines the individual record length.

offset	Offset of the record length field from the start of the record.
length	Length of the binary record length field.
origin	The start of the record data at which the record length is applied. Default is zero

◊ *STD|NL|CR|LF|CRLF|LFCR|CRNL|**string***
Sets the input end-of-line delimiter value used to determine the end of each record for non-RECFM F or V input. Delimiters are not included in the edited record data or record length. Parameter elements are as follow:

STD		See below.
NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
string	-	A 2-byte user specified character or hex string. e.g. c'ZZ' X'ABCD'

Default is *STD*, meaning that SELCOPY/i will attempt to automatically identify the end-of-line character(s) in use used by scanning for the first occurrence of one of the following standard EOL combinations: CR, LF, NL, CRNL, CRLF or LFCR.

Lrecl:

For HFS only, specifies the maximum record length of input records.

Records terminated by an EOL sequence will wrap onto the next line of data if the record length exceeds *lrecl*. Where a record has wrapped, the prefix area contains the "==EOL>" flag. Furthermore, read-only edit is forced in order to suppress save of a wrapped record as multiple, individual records.

For RECFM F data, *lrecl* is the fixed length of the records in the edit view. If the file size is not a multiple of the fixed format lrecl value then, an error occurs and edit is cancelled. For display purposes only, using BROWSE with any *lrecl* value will display the data with the last record padded with blanks up to the *lrecl* length.

If the record length field of a RECFM V record exceeds the *lrecl* value, then an error is returned.

RECFM V and EOL delimited records have default *lrecl* of 32752, wheras RECFM F records have default *lrecl* of 80.

`Browse`

Indicates that the datset will be opened for Browse. The Edit and Browse options are mutually exclusive.

`Edit`

Indicates that the datset will be opened for Edit. The Edit and Browse options are mutually exclusive.

If Edit is selected, then the type of edit to be used may also be specified as follows:

Allow Ins/Del	Perform full function edit of the data. Records may be inserted, deleted, copied, moved and the record contents altered so that, if appropriate, the record length is changed. This is default if no edit type is specified. For non-KSDS files that are too large to be read complete into available storage, this type of edit requires that an auxiliary copy of the whole file is taken before the edit session can continue. SELCOPY/i optimises performance and storage requirements when editing large files, by keeping in storage only those records that have changed, or are required to display any SDE views of the file.
Upd-In-Place	Perform Update in place edit of the data. Records can not be inserted, deleted, copied or moved. Record contents may still be altered, however, record length must not change. For this type of edit, records are read from the file only as it becomes necessary to display them.
Read-Only	Perform full function edit of the data as detailed in "Allow Ins/Del", however, changes may not be saved unless the fileid is first updated to that of a non-existant data set. (See SDE CLI command, SET FILEID or any of the alternative SDE SET options that alter the fileid.) On saving the data, the user will be prompted to allocate/define the new data set.

Each of these edit type fields are mutually exclusive.

`FROM:`

If activated, the FROM field specifies a record location value identifying the record within the data set to be displayed first (record 1) in the the SDE browse/edit window view. Records that occur before this record are not included within the browse/edit session.

If FILTER is specified, then record filtering occurs only on records selected using the FROM and FOR parameters.

Note that, if FROM is specified for Edit, then **Upd-In-Place** edit must be selected.

Default is to display records within the SDE window view starting at the first record in the data set.

At least one of the mutually exclusive options **Record**, **Key** or **RBA** must be selected to identify the format of the record location.

If record location is by Record or RBA, then the FROM value may be specified as an integer numeric value (e.g. **123**) or as a hexadecimal numeric value (e.g. **X'7B'**).

If record location is by Key, then the FROM value may be specified as a string literal (e.g. **abc** or **'abc'**), which will be upper cased before keyed look-up; as a character string (e.g. **C'abc'**), which preserves character case; or a hexadecimal string (e.g. **X'818283'**).

`Record`

Record may be selected for files of any data set organisation and indicates that the **FROM** field represents a **record number** within the data set.

`Key`

Valid only for VSAM KSDS, **Key** indicates that the **FROM** field represents a KSDS key field string. If the key is not found, the next record with the a key string greater that that specified, becomes the first record of the display.

`RBA`

Valid only for VSAM ESDS, **RBA** indicates that the **FROM** field represents a relative byte address within the data set. The RBA location must point to the start of a record otherwise a VSAM point error will occur.

RBA address for each record is displayed in the SDE view of the file records as part of the record information columns within an SDE window view using the SDE command, SET RECINFO.

`FOR:`

If activated, the FOR field specifies the maximum total number of records to be available for display within the SDE window view. (i.e. the End-of-Data line occurs following this number of records).

Default is to make available all records following the **FROM** start location.

Note that if a **FOR** field value is used, then enough available storage must exist in order to load all selected records. Furthermore, if FOR is specified for Edit, then **Upd-In-Place** edit must be selected.

When used in conjunction with a FILTER, the FOR value limits the number of records read from the file when attempting to satisfy the filter's selection criteria.

FILTER:

If activated, the FILTER field specifies either a filter clause or the fileid of a saved filter file.

Note that the FILTER menu item may be selected to build and save a file filter file using the Create File Filter panels.

Use a filter to restrict records loaded into storage to only those that match the selection criteria specified by the filter. Note that if a filter is specified, then enough available storage must exist in order to load all selected records.

A FILTER may be used in conjunction with a FOR parameter in order to limit the number of records read from the file when attempting to satisfy the filter's selection criteria.

For filter clause syntax, see description of the SDE EDIT command Filter Clause.

USING (SDO) Structure:

If activated, The USING SDO field specifies the name of an existing SDE structure (SDO) to be used to map record fields in the edited data set.

An SDO may be created from COBOL or PL1 copybooks; COBOL or PL1 ADATA compile output files; SELCOPY/i SDE CREATE STRUCTURE syntax or and XREF file.

Use panel Create Structure from COBOL/PL1 Copybook(s), opened on selecting **Structure** from the Primary Option Menu

If no structure is specified, each data set record will be of the default record type "Unmapped", i.e. a single character field of length equal to that of the record.

USING *struct_type* Copybook:

If activated, the USING Copybook field specifies the name of a COBOL or PL1 copybook, or COBOL or PL1 ADATA dataset name, to be used to generate a temporary SDE structure (SDO) necessary for the structured browse/edit.

For COBOL or PL1 copybooks the language type (COBOL or PL1) should be entered in the field following USING. Alternatively, if COBOL or PL1 ADATA output is to be used, ADATA should be entered instead.

The generated SDO is used to map records in the data set for browse or edit.
Note that generating an SDO before each browse or edit has a performance overhead. Therefore, it is recommended that a permanent copy of the SDO be created on disk and then this used to map the record data.

If menu item SDO is selected, then these input fields, together with the specified USING SDO field data set name, are used to generate a non-temporary SDE structure (SDO). Note, however, that panel Create Structure from COBOL/PL1 Copybook(s) offers better flexibility when generating an SDO from COBOL or PL1 copybooks.

Default Record-Type:

If activated, the Default Record-Type field is used in conjunction with the USING Copybook field to specify a record type name to be used when no COBOL LEVEL 01 or PL1 major structure name exists in the specified copybook.

For more information on use of COBOL and PL1 copybooks where no level 01 record exists, refer to CREATE STRUCTURE.

Window Rows: *n_rows* x Columns: *n_cols*

If SELCOPY/i windows are not in a maximised state, these fields specify the required dimensions (rows by columns) of the resulting SDE browse/edit window view.

List File Windows (=3)

The List Menu panel (ZZSGLIST) is an interactive panel window opened on selection of option 3. in the SELCOPY/i Primary option menu.

SELCOPY/i file lists provide detailed information for DASD files and related system resources. (e.g. ENQs, DASD and Associated Cataloged objects.)

All list file windows are of window class, LISTFILE, and have common characteristics defined for list window classes.

Options

1 Volumes	LVOL - List DASD Volumes
2 VTOC	LV - List VTOC files
3 Extents	LX - List VTOC Extents
4 Catalog	LC - List Cataloged datasets (catalog detail)
5 Dataset	LD - List Dataset details (catalog & VTOC detail)
6 Library	LL - List PDS/PDSE Library members
7 Allocated	LA - List Allocated files (DD names)
8 Enqueues	LQ - List Resource Enqueues
9 Job Enqueues	LJQ - List Job Enqueues
10 Associations	LAS - List File Assocations
11 HFS	LP - List HFS Paths
12 DB2	LDxx - List DB2 objects

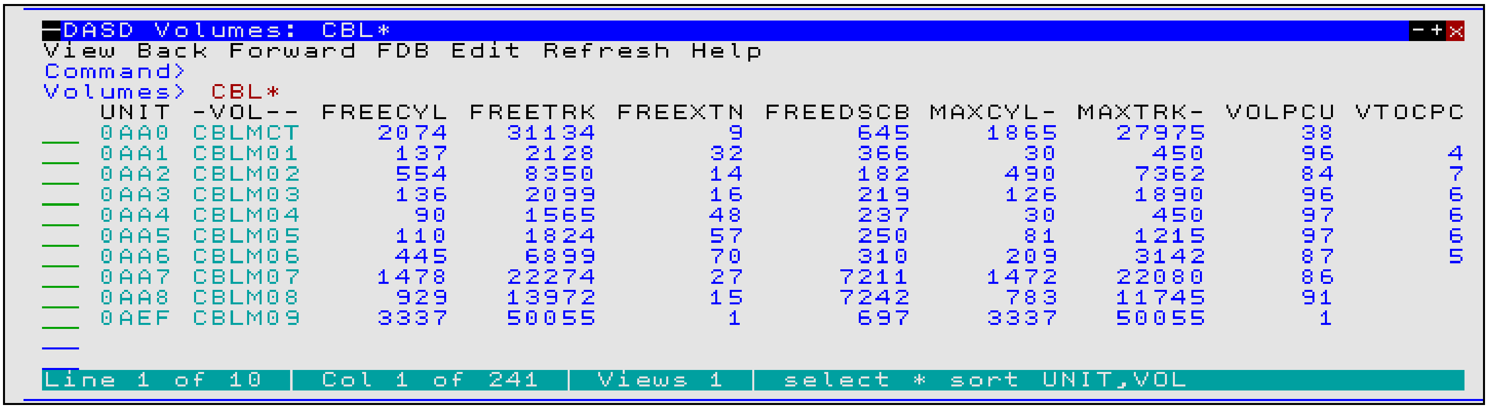
List DASD Volumes (=3.1)

See List Windows for general features and commands common to all list windows.

The DASD Volumes List window may be opened via the following:

- Select option 1. 'Volumes' from the List Menu.
- Select 'DASD Volumes' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LVOL on the command line of any window.

The DASD Volumes window displays the attributes of DASD volumes defined to your system.



```
DASD Volumes: CBL*                                                          -+x
View Back Forward FDB Edit Refresh Help
Command>
Volumes>  CBL*
    UNIT -VOL-- FREECYL FREETRK FREEXTN FREEDSCB MAXCYL- MAXTRK- VOLPCU VTOCPC
___ 0AA0 CBLMCT    2074   31134       9      645    1865   27975     38
___ 0AA1 CBLM01     137    2128      32      366      30     450     96      4
___ 0AA2 CBLM02     554    8350      14      182     490    7362     84      7
___ 0AA3 CBLM03     136    2099      16      219     126    1890     96      6
___ 0AA4 CBLM04      90    1565      48      237      30     450     97      6
___ 0AA5 CBLM05     110    1824      57      250      81    1215     97      6
___ 0AA6 CBLM06     445    6899      70      310     209    3142     87      5
___ 0AA7 CBLM07    1478   22274      27     7211    1472   22080     86
___ 0AA8 CBLM08     929   13972      15     7242     783   11745     91
___ 0AEF CBLM09    3337   50055       1      697    3337   50055      1
___
Line 1 of 10 | Col 1 of 241 | Views 1 | select * sort UNIT,VOL
```

Figure 11. DASD Volumes window.

Panel Input Fields

`Volume>`

Specify a volume id mask. The mask supports the following wild cards:

`*` An asterisk indicates that one or more characters within the volume id can occupy that position. An asterisk can precede or follow a set of characters.

`%` A single percent sign indicates that exactly one character can occupy that position. (Up to 6 percent signs can be specified.)

By default, a volume id mask that is less than 6 characters in length and does not contain an * (asterisk) wild card will be treated as having an implied trailing * wild card.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix Line command T.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
T	Open the VTOC list window for the volume.
VC	Open an Execute CBLVCAT window and issue a LISTVTOC operation for the entry.
?	Open the volume statistics window for the volume containing the file.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

The data displayed for MVS is:

Name	Type	Description
UNIT	Hex	Unit address
VOL	Char	Volume serial number
FREECYL	UInt	Free cylinders
FREETRK	UInt	Free tracks
FREEXTN	UInt	Free extents
FREEDSCB	UInt	Free DSCBs
MAXCYL	UInt	Largest free extent (CYLs)
MAXTRK	UInt	Largest free extent (TRKs)
VOLPCU	UInt	Volume percent used
VTOCPCU	UInt	VTOC percent used
TOTALCYL	UInt	Total cylinders
TRKCYL	UInt	Tracks per cylinder
TRKLEN	UInt	Track length
UCBTYPE	Hex	Unit type
SMS	Char	SMS managed indicator
VTOCIX	Char	Indexed VTOC
VTOCXTN	UInt	Number of VTOC extents
VTOCTRK	UInt	Number of VTOC tracks
LOWCCHH	Hex	VTOC start CCHH
HIGHCCHH	Hex	VTOC end CCHH
DSCBTRK	UInt	DSCBs per track
FREEVIR	UInt	Number of free VTOC index recs
FRAGINDX	UInt	Fragmentation index
ALTCCHH	Hex	Next available alt track CCHH
ALTREM	UInt	Remaining alternate tracks
MOUNT	Char	Mount usage status
DEV	Char	Device type
MODEL	Char	Device model

MODELX	Hex	Device model (hex)
CACHE	Char	Cached device
SHARE	Char	Shareable device

The data displayed for VSE is:

Name	**Type**	**Description**
UNIT	Hex	Unit address
VOL	Char	Volume serial number
TYPE	Char	External device type code
FORMAT	Hex	Device format
AVRVTOC	Hex	VTOC address
PUBC	Hex	PUB device type code
DTFC	Hex	DTF device type code
UCBC	Hex	Unit code
DCTPCYL	UInt	Primary cylinders
DCTACYL	UInt	Alternate cylinders
DCTTCYL	UInt	Tracks per cylinder
DCTBTRK	UInt	Bytes per track
DCTTFIX	UInt	Cylinders under fixed head
DCTMAXR	UInt	Maximum physical record size
DCTDEVC	Hex	Device constants

List VTOC Files (=3.2)

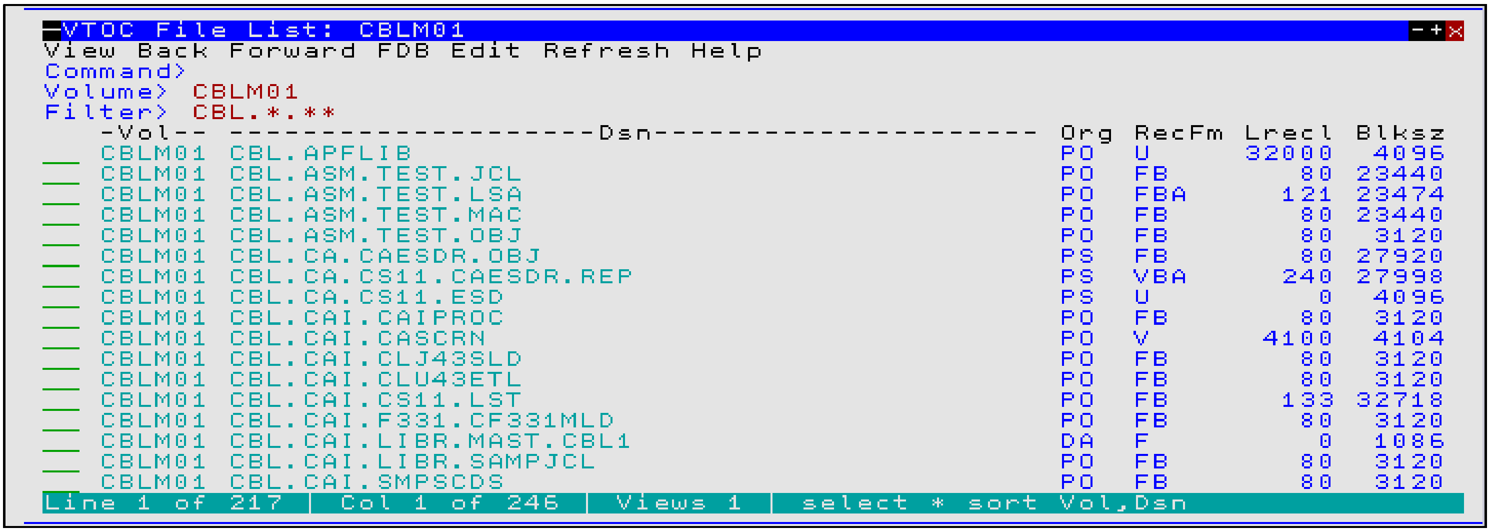
See List Windows for general features and commands common to all list windows.

The VTOC File List window may be opened via the following:

- Select option 2. 'VTOC' from the List Menu.
- Select 'VTOC Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LV on the command line of any window.

The VTOC File List window displays data set entry information in a DASD volume's Volume Table of Contents (VTOC).

Note: List VTOC Files is not supported for CMS.



```
VTOC File List: CBLM01
View Back Forward FDB Edit Refresh Help
Command>
Volume> CBLM01
Filter> CBL.*.**
   -Vol-- --------------------Dsn---------------------- Org RecFm Lrecl Blksz
___ CBLM01 CBL.APFLIB                                   PO  U     32000  4096
___ CBLM01 CBL.ASM.TEST.JCL                             PO  FB       80 23440
___ CBLM01 CBL.ASM.TEST.LSA                             PO  FBA     121 23474
___ CBLM01 CBL.ASM.TEST.MAC                             PO  FB       80 23440
___ CBLM01 CBL.ASM.TEST.OBJ                             PO  FB       80  3120
___ CBLM01 CBL.CA.CAESDR.OBJ                            PS  FB       80 27920
___ CBLM01 CBL.CA.CS11.CAESDR.REP                       PS  VBA     240 27998
___ CBLM01 CBL.CA.CS11.ESD                              PS  U         0  4096
___ CBLM01 CBL.CAI.CAIPROC                              PO  FB       80  3120
___ CBLM01 CBL.CAI.CASCRN                               PO  V      4100  4104
___ CBLM01 CBL.CAI.CLJ43SLD                             PO  FB       80  3120
___ CBLM01 CBL.CAI.CLU43ETL                             PO  FB       80  3120
___ CBLM01 CBL.CAI.CS11.LST                             PO  FB      133 32718
___ CBLM01 CBL.CAI.F331.CF331MLD                        PO  FB       80  3120
___ CBLM01 CBL.CAI.LIBR.MAST.CBL1                       DA  F         0  1086
___ CBLM01 CBL.CAI.LIBR.SAMPJCL                         PO  FB       80  3120
___ CBLM01 CBL.CAI.SMPSCDS                              PO  FB       80  3120
Line 1 of 217 | Col 1 of 246 | Views 1 | select * sort Vol,Dsn
```

Figure 12. VTOC File List window displaying all entries beginning 'CBL.' on volume 'CBLM01.'

Panel Input Fields

Volume>

The 1-6 character volume id containing the required VTOC.

Filter>

Note: The filter parameter is not supported for VSE.

Select only data sets that match the specified filter mask. The filter mask supports the following wild cards:

*	A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
**	A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
%	A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

A filter field that contains **neither** "*" (asterisk) nor "**" (double asterisk) wild cards will have a wildcard string of "*.**" automatically appended and so list all those data sets whose names begin with the filter string.

Prefix Line Commands

The following prefix line commands are available:

Command	**Description**
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	For MVS only, open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS(E), open the File Search window for the entry.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	For MVS only, list dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.

>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
Vol	Char	Volume serial number
Dsn	Char	Dataset name
Org	Enum	Data set organisation
RecFm	Enum	Record format
Lrecl	UInt	Logical record length
Blksz	UInt	Block Size
Alu	Char	Allocation unit
Pri	UInt	Primary space allocation
Sec	UInt	Secondary space allocation
Alt	UInt	Allocation total
Nxt	UInt	Number of extents
Trks	UInt	Tracks allocated
DsnPcu	UInt	Dataset percent used
DsKb	UInt	Dataset space Kilobytes
Created	VTOCDate	Creation date
Referenced	VTOCDate	Last referenced date
Expires	VTOCDate	Expiry Date
SMS	BitFlag	SMS managed dataset
PDSE	BitFlag	PDS extended
HFS	BitFlag	HFS dataset
UnCat	BitFlag	Uncataloged dataset
XFD	BitFlag	Extended format dataset
XAttr	BitFlag	Extended attributes exist
ReBlk	BitFlag	May be reblocked
IsICF	BitFlag	ICF catalog BCS dataset
InICF	BitFlag	Cataloged in an ICF catalog
DSInd	Hex	Dataset indicators (DS1DSIND)
KyL	UInt	Dataset key length
RKP	UInt	Relative key position
TBal	UInt	Bytes remaining on last track
BlKTrk	UInt	Blocks per track
F1Vol	Char	Format 1 DSCB volume serial
VSeq	UInt	Volume sequence number

List VTOC Extents (=3.3)

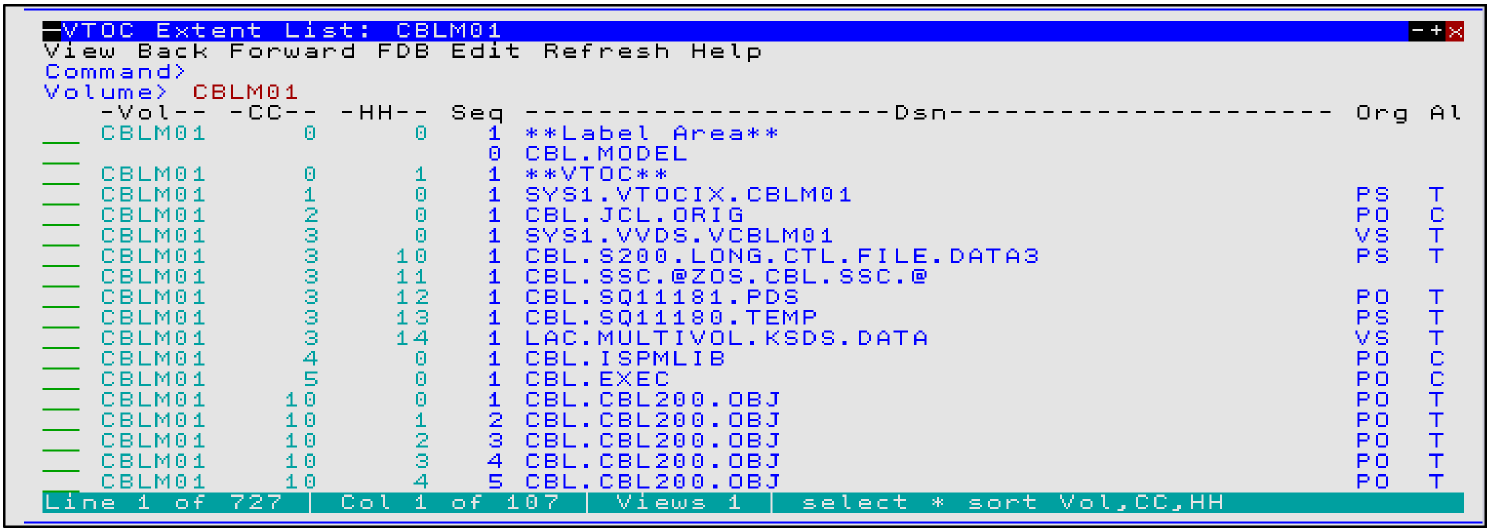
See List Windows for general features and commands common to all list windows.

The VTOC Extent List window may be opened via the following:

- Select option 3. 'Extents' from the List Menu.
- Select 'VTOC Extents' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LX on the command line of any window.

The VTOC Extent List window displays all information in a DASD volume's Volume Table of Contents (VTOC) by physical extent. This includes free extents and volume control areas such as the VTOC and the label area.

Note: Not supported for VSE.



```
 VTOC Extent List: CBLM01                                                    -+x
View Back Forward FDB Edit Refresh Help
Command>
Volume> CBLM01
   -Vol-- -CC-- -HH-- Seq --------------------Dsn--------------------- Org Al
___ CBLM01     0     0   1 **Label Area**
___                      0 CBL.MODEL
___ CBLM01     0     1   1 **VTOC**
___ CBLM01     1     0   1 SYS1.VTOCIX.CBLM01                           PS  T
___ CBLM01     2     0   1 CBL.JCL.ORIG                                 PO  C
___ CBLM01     3     0   1 SYS1.VVDS.VCBLM01                            VS  T
___ CBLM01     3    10   1 CBL.S200.LONG.CTL.FILE.DATA3                 PS  T
___ CBLM01     3    11   1 CBL.SSC.@ZOS.CBL.SSC.@
___ CBLM01     3    12   1 CBL.SQ11181.PDS                              PO  T
___ CBLM01     3    13   1 CBL.SQ11180.TEMP                             PS  T
___ CBLM01     3    14   1 LAC.MULTIVOL.KSDS.DATA                       VS  T
___ CBLM01     4     0   1 CBL.ISPMLIB                                  PO  C
___ CBLM01     5     0   1 CBL.EXEC                                     PO  C
___ CBLM01    10     0   1 CBL.CBL200.OBJ                               PO  T
___ CBLM01    10     1   2 CBL.CBL200.OBJ                               PO  T
___ CBLM01    10     2   3 CBL.CBL200.OBJ                               PO  T
___ CBLM01    10     3   4 CBL.CBL200.OBJ                               PO  T
___ CBLM01    10     4   5 CBL.CBL200.OBJ                               PO  T
Line 1 of 727 | Col 1 of 107 | Views 1 | select * sort Vol,CC,HH
```

Figure 13. VTOC Extent List window displaying all extents on volume 'CBLM01.'

Panel Input Fields

Volume>
The 1-6 character volume id containing the required VTOC.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS(E), open the File Search window for the entry.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.

UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	**Type**	**Description**
Vol	Char	Volume serial number
CC	UInt	Cylinder number (decimal)
HH	UInt	Head number (decimal)
Seq	UInt	Extent sequence
Dsn	Char	Dataset name
Org	Enum	Data set organisation
Alu	Char	Allocation unit
Trks	UInt	Tracks allocated
Nxt	UInt	Number of extents
LoCCHH	Hex	Extent Low CCHH
HiCCHH	Hex	Extent High CCHH

List Catalog Entries (=3.4)

See List Windows for general features and commands common to all list windows.

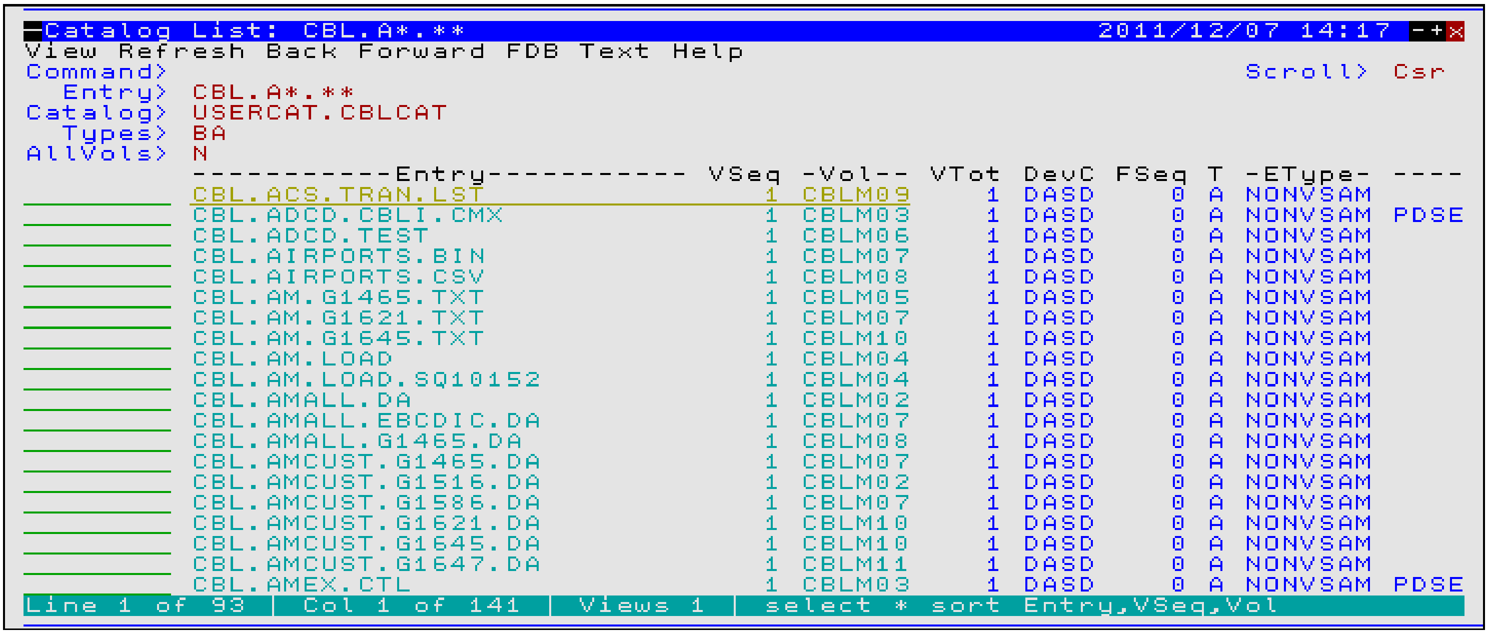
The Catalog List window may be opened via the following:

- Select option 4. 'Catalog' from the List Menu.
- Select 'Cataloged Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LC on the command line of any window.

For CMS, the File List window is opened in place of the Catalog List or Dataset List window and displays information about files residing on accessed mini-disks. Refer to documentation on List CMS Files.

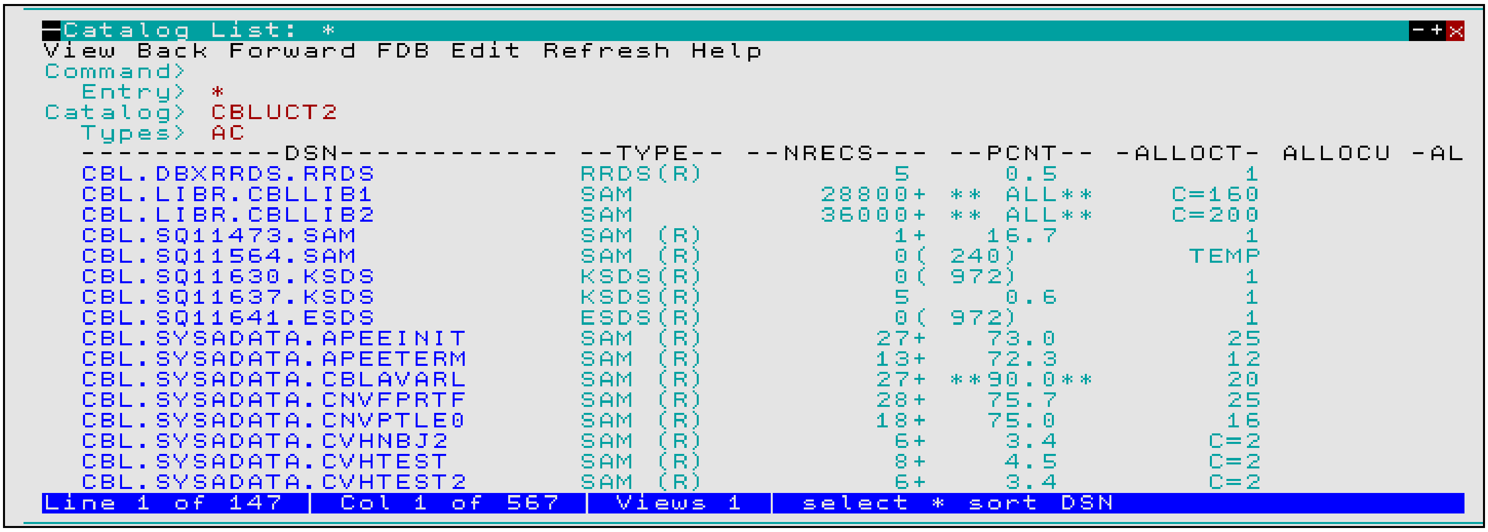
For VSE, the Catalog List window is supported only where the CBL software product CBLVCAT is installed and active. The Catalog List window uses CBLVCAT to read the specified VSAM catalog records to obtain information about the cataloged files.

For MVS, the Catalog List window displays the basic catalog entry information for ICF cataloged data sets. The Dataset List window should be used to display more detailed information on cataloged data sets.



```
 Catalog List: CBL.A*.**                                       2011/12/07 14:17 -+x
View Refresh Back Forward FDB Text Help
Command>                                                            Scroll> Csr
   Entry> CBL.A*.**
 Catalog> USERCAT.CBLCAT
   Types> BA
 AllVols> N
          -----------Entry----------- VSeq -Vol-- VTot DevC FSeq T -EType- ----
          CBL.ACS.TRAN.LST               1 CBLM09    1 DASD    0 A NONVSAM
          CBL.ADCD.CBLI.CMX              1 CBLM03    1 DASD    0 A NONVSAM PDSE
          CBL.ADCD.TEST                  1 CBLM06    1 DASD    0 A NONVSAM
          CBL.AIRPORTS.BIN               1 CBLM07    1 DASD    0 A NONVSAM
          CBL.AIRPORTS.CSV               1 CBLM08    1 DASD    0 A NONVSAM
          CBL.AM.G1465.TXT               1 CBLM05    1 DASD    0 A NONVSAM
          CBL.AM.G1621.TXT               1 CBLM07    1 DASD    0 A NONVSAM
          CBL.AM.G1645.TXT               1 CBLM10    1 DASD    0 A NONVSAM
          CBL.AM.LOAD                    1 CBLM04    1 DASD    0 A NONVSAM
          CBL.AM.LOAD.SQ10152            1 CBLM04    1 DASD    0 A NONVSAM
          CBL.AMALL.DA                   1 CBLM02    1 DASD    0 A NONVSAM
          CBL.AMALL.EBCDIC.DA            1 CBLM07    1 DASD    0 A NONVSAM
          CBL.AMALL.G1465.DA             1 CBLM08    1 DASD    0 A NONVSAM
          CBL.AMCUST.G1465.DA            1 CBLM07    1 DASD    0 A NONVSAM
          CBL.AMCUST.G1516.DA            1 CBLM02    1 DASD    0 A NONVSAM
          CBL.AMCUST.G1586.DA            1 CBLM07    1 DASD    0 A NONVSAM
          CBL.AMCUST.G1621.DA            1 CBLM10    1 DASD    0 A NONVSAM
          CBL.AMCUST.G1645.DA            1 CBLM10    1 DASD    0 A NONVSAM
          CBL.AMCUST.G1647.DA            1 CBLM11    1 DASD    0 A NONVSAM
          CBL.AMEX.CTL                   1 CBLM03    1 DASD    0 A NONVSAM PDSE
Line 1 of 93 | Col 1 of 141 | Views 1 | select * sort Entry,VSeq,Vol
```

Figure 14. MVS Catalog List Window.



```
 Catalog List: *                                                           -+x
View Back Forward FDB Edit Refresh Help
Command>
   Entry> *
 Catalog> CBLUCT2
   Types> AC
   -----------DSN------------ --TYPE-- --NRECS--- --PCNT-- -ALLOCT- ALLOCU -AL
   CBL.DBXRRDS.RRDS           RRDS(R)           5      0.5        1
   CBL.LIBR.CBLLIB1           SAM          28800+  ** ALL**   C=160
   CBL.LIBR.CBLLIB2           SAM          36000+  ** ALL**   C=200
   CBL.SQ11473.SAM            SAM (R)          1+     16.7        1
   CBL.SQ11564.SAM            SAM (R)          0(  240)        TEMP
   CBL.SQ11630.KSDS           KSDS(R)          0(  972)           1
   CBL.SQ11637.KSDS           KSDS(R)          5       0.6        1
   CBL.SQ11641.ESDS           ESDS(R)          0(  972)           1
   CBL.SYSADATA.APEEINIT      SAM (R)         27+     73.0       25
   CBL.SYSADATA.APEETERM      SAM (R)         13+     72.3       12
   CBL.SYSADATA.CBLAVARL      SAM (R)         27+  **90.0**      20
   CBL.SYSADATA.CNVFPRTF      SAM (R)         28+     75.7       25
   CBL.SYSADATA.CNVPTLE0      SAM (R)         18+     75.0       16
   CBL.SYSADATA.CVHNBJ2       SAM (R)          6+      3.4      C=2
   CBL.SYSADATA.CVHTEST       SAM (R)          8+      4.5      C=2
   CBL.SYSADATA.CVHTEST2      SAM (R)          6+      3.4      C=2
Line 1 of 147 | Col 1 of 567 | Views 1 | select * sort DSN
```

Figure 15. VSE Catalog List Window.

Panel Input Fields

`Entry>`

Specify the fileid mask.

◊ For **MVS** systems, the fileid mask represents a DSN mask that supports the following wild cards:

`*` A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.

`**` A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.

`%` A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                   becomes:   DEV*
DEV.OEM.TRSPAN*.        becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.          becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                    becomes:   DEV.**
DEV*                   becomes:   DEV*.**
DEV.OEM.TRSPAN*        becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*            becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN         becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%            becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19             becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

◊ For **VSE** systems, the fileid mask is a valid CBLVCAT LISTCAT KEY parameter string. i.e. entries with file name **beginning** with the specified string or, if prefixed by "/" (slash), entries with file name **containing** the specified string. (See the CBLVCAT User Manual.)

If no fileid mask is specified, all entries will be selected.
Note that wild cards are not supported within the VSE fileid mask, however, "*" (asterisk) is tolerated if placed at the end of the fileid mask.

Catalog>

Nominate a specific catalog in which to search for the requested entry.

For **MVS** systems, this is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> field.

For **VSE** systems, this is a disk label assigned to the VSAM catalog for which entries are to be listed.
If no catalog file label is specified, the Catalog List window displays all user catalogs cataloged in the master catalog.

Default is the master catalog.

Types>

Specify the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

AllVols>

Specify "Y" or "N" to control whether repeated display of the same entry occurs for multi-volume data sets. If "N" is specified, then only the primary volume entry is displayed.

Prefix Line Commands

For **MVS** systems, the following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.

CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	Open the File Search window for the entry.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

For **VSE** systems, the following prefix line commands are available:

Command	**Description**
D	Delete the entry. User will be prompted to verify the deletion.
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

For **MVS** systems, the data displayed is:

Name	**Type**	**Description**
Entry	Char	Entry name
Vol	Char	Volume serial number
VSeq	UInt	Volume Sequence number
DevC	Char	Device Class
FSeq	UInt	File Sequence number

T	Char	Catalog entry type code
EType	Char	Catalog entry type
DSType	Char	Dataset type
UnitType	Hex	Unit type
UnitName	Char	Unit name
DataClas	Char	SMS Data Class
StorClas	Char	SMS Storage Class
MgmtClas	Char	SMS Management Class

For **VSE** systems, the data displayed is:

Name	Type	Description
ALLOCP	Char	Defined Primary Allocation
ALLOCS	Char	Defined Secondary Allocation
ALLOCT	Char	Current Total Space Allocation
ALLOCU	Char	Unused Allocated Space
AVRL	Char	Defined Average Record Length
BLKSIZE	Char	Defined VSAM SAM Block Size
BUFSP	Char	Defined Buffer Space (BUFSP)
BUFSP/IXL	Char	Defined BUFSP or INDEX levels
CATALOG	Char	Catalog File Name
CI/CA	Char	Number of Control Intervals/Control Area
CISIZE	Char	Defined Control Interval Size
COMPONENT	Char	VSAM Object Component Name
DEFINED	Char	Date the File was Defined
DSN	Char	Fileid
ENTRY	Char	VSAM Component Entry Name
EXCPS	Char	Number of Executed Channel Programs
EXPIRES	Char	Expiry Date
FREEBYTES	Char	Free Space Bytes Value
FRSP	Char	Defined Freespace (Bytes/CI and CI/CA)
HIALLRBA	Char	Current High Allocated RBA
HIUSERBA	Char	Current High Used RBA
IMB/REP	Char	Defined Index Attributes (IMBED and/or REPLICATE)
IXL	Char	Number of INDEX Levels
KL	Char	Defined KEY Length
KL/BLK/IMB	Char	Merge KL, BLK and IMB/REP Values
LMAX	Char	Defined Maximum Record Length
NRECS	Char	Current Number of Records
NSEC	Char	Number of Allocated Extents Minus One
PCNT	Char	Calculated Amount of Used Space
PHYREC	Char	Physical Record Size allocated by VSAM
RECSTATS	Char	Number of Records Deleted, Inserted, Updated and Read
RKP	Char	Defined Relative KEY Position
S/C	Char	Defined Local Share option and the Primary Space Class
SEVL	Char	CBLVCAT's Highest Severity Level Massage Reference for the File
SHR	Char	Defined Local (Cross Region) and Cross System Share Options
SPLITCA	Char	Number of CA Splits to Date.

SPLITCI	Char	Number of CI Splits to Date.
TIMESTMP	Char	Time Stamp of VSAM object (Last Closed)
TYPE	Char	File Type
VOLUME	Char	Defined Primary Volume Serial Number

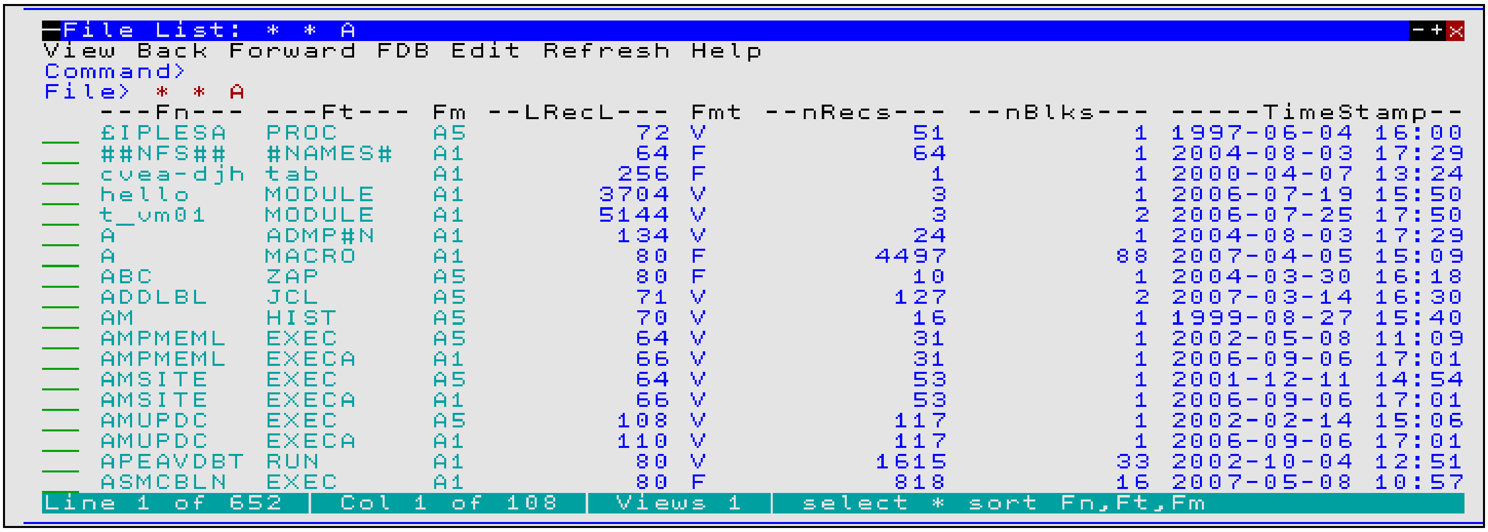
List CMS Files

See List Windows for general features and commands common to all list windows.

Supported for CMS systems only, the File List window may be opened via the following:

- Enter command FL (synonym for LC) or LD on the command line of any window.

For CMS, the **File List** window is opened in place of the MVS/VSE Catalog List or Dataset List windows and displays information about files residing on accessed disks.



```
 File List: * * A
View Back Forward FDB Edit Refresh Help
Command>
File> * * A
    ---Fn---  ---Ft---  Fm --LRecL--- Fmt --nRecs--- --nBlks--- -----TimeStamp--
___ £IPLESA   PROC      A5         72 V           51          1 1997-06-04 16:00
___ ##NFS##   #NAMES#   A1         64 F           64          1 2004-08-03 17:29
___ cvea-djh  tab       A1        256 F            1          1 2000-04-07 13:24
___ hello     MODULE    A1       3704 V            3          1 2006-07-19 15:50
___ t_vm01    MODULE    A1       5144 V            3          2 2006-07-25 17:50
___ A         ADMP#N    A1        134 V           24          1 2004-08-03 17:29
___ A         MACRO     A1         80 F         4497         88 2007-04-05 15:09
___ ABC       ZAP       A5         80 F           10          1 2004-03-30 16:18
___ ADDLBL    JCL       A5         71 V          127          2 2007-03-14 16:30
___ AM        HIST      A5         70 V           16          1 1999-08-27 15:40
___ AMPMEML   EXEC      A5         64 V           31          1 2002-05-08 11:09
___ AMPMEML   EXECA     A1         66 V           31          1 2006-09-06 17:01
___ AMSITE    EXEC      A5         64 V           53          1 2001-12-11 14:54
___ AMSITE    EXECA     A1         66 V           53          1 2006-09-06 17:01
___ AMUPDC    EXEC      A5        108 V          117          1 2002-02-14 15:06
___ AMUPDC    EXECA     A1        110 V          117          1 2006-09-06 17:01
___ APEAVDBT  RUN       A1         80 V         1615         33 2002-10-04 12:51
___ ASMCBLN   EXEC      A1         80 F          818         16 2007-05-08 10:57
Line 1 of 652 | Col 1 of 108 | Views 1 | select * sort Fn,Ft,Fm
```

Figure 16. CMS File List window displaying all files on mini-disk A.

Panel Input Fields

`File>`

Specify the CMS fileid mask.

The fileid mask may consist of up to 3 qualifiers representing a filename filetype filemode combination where qualifiers are separated by one or more blanks or a "." (dot/period).

A single "*" (asterisk) wild card may be used to represent an entire qualifier or zero or more characters at a particular position within the qualifier. Wild card "*" may be specified more that once, anywhere within a qualifier.

Default filemode qualifier is "A", default filetype qualifier is "*".

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
C	Copy the entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
F	Open the file search window for the PDS.
K	Delete (Kill) the entry without prompting for verification.

R	Rename the entry.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	**Type**	**Description**
nBlks	UInt	Number of blocks.
nRecs	UInt	Number of records.
Entry	Char	File id.
Fm	Char	File mode.
Fmt	Char	Record format.
Fn	Char	File name.
Ft	Char	File type
Label	Char	Disk label.
LRecL	UInt	Record length.
TimeStamp	Char	Last update date and time.

List Dataset Details (=3.5)

See List Windows for general features and commands common to all list windows.

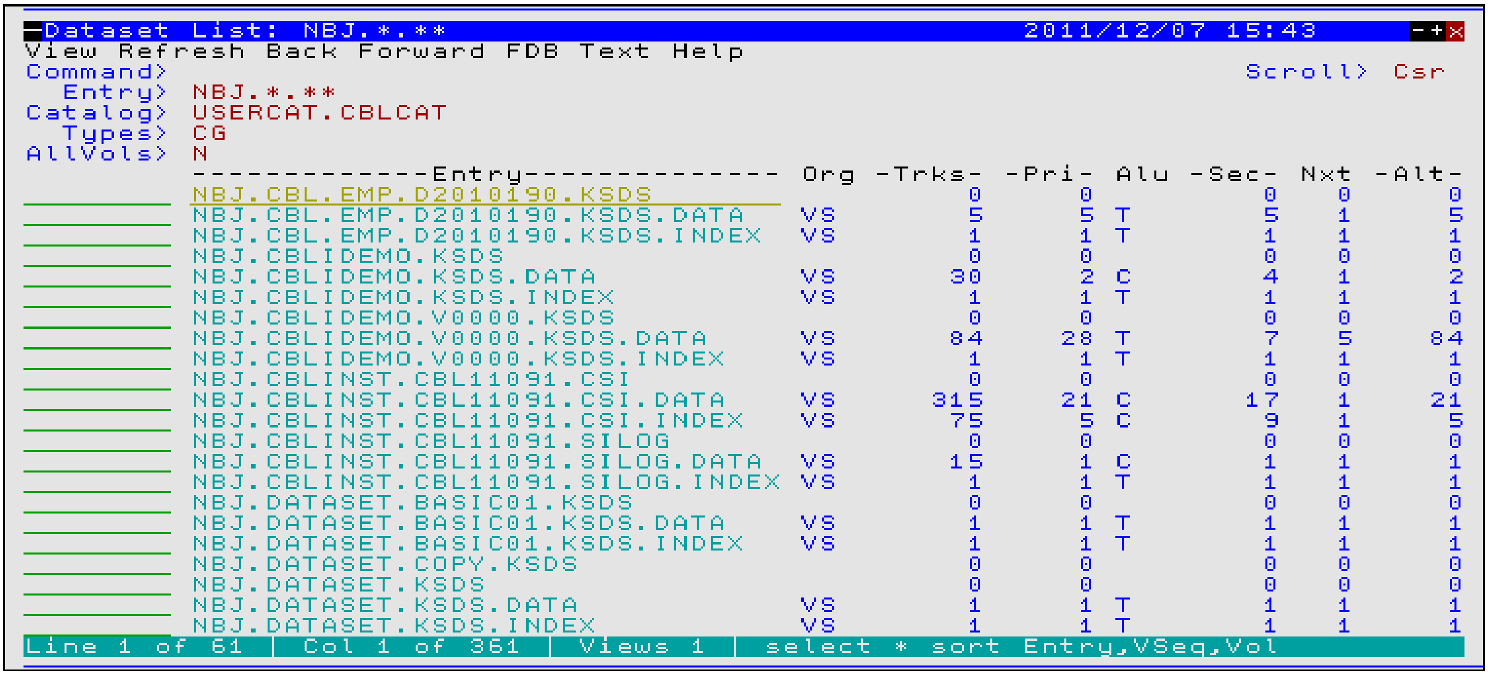
The Dataset List window may be opened via the following:

- Select option 5. 'Dataset' from the List Menu.
- Select 'Dataset Details' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LD on the command line of any window.

For CMS, the File List window is opened in place of the Catalog List or Dataset List window and displays information about files residing on accessed mini-disks. Refer to documentation on List CMS Files.

The Dataset List window is not supported on **VSE** systems.

The Dataset List window displays the basic catalog entry information together with the details of their geometry obtained either from the catalog or the VTOC for cataloged data sets.



```
 Dataset List: NBJ.*.**                                    2011/12/07 15:43   -+x
View Refresh Back Forward FDB Text Help
Command>                                                             Scroll> Csr
  Entry> NBJ.*.**
Catalog> USERCAT.CBLCAT
  Types> CG
AllVols> N
         -------------Entry--------------- Org -Trks- -Pri- Alu -Sec- Nxt -Alt-
         NBJ.CBL.EMP.D2010190.KSDS                0     0         0   0     0
         NBJ.CBL.EMP.D2010190.KSDS.DATA    VS      5     5 T       5   1     5
         NBJ.CBL.EMP.D2010190.KSDS.INDEX   VS      1     1 T       1   1     1
         NBJ.CBLIDEMO.KSDS                        0     0         0   0     0
         NBJ.CBLIDEMO.KSDS.DATA            VS     30     2 C       4   1     2
         NBJ.CBLIDEMO.KSDS.INDEX           VS      1     1 T       1   1     1
         NBJ.CBLIDEMO.V0000.KSDS                  0     0         0   0     0
         NBJ.CBLIDEMO.V0000.KSDS.DATA      VS     84    28 T       7   5    84
         NBJ.CBLIDEMO.V0000.KSDS.INDEX     VS      1     1 T       1   1     1
         NBJ.CBLINST.CBL11091.CSI                 0     0         0   0     0
         NBJ.CBLINST.CBL11091.CSI.DATA     VS    315    21 C      17   1    21
         NBJ.CBLINST.CBL11091.CSI.INDEX    VS     75     5 C       9   1     5
         NBJ.CBLINST.CBL11091.SILOG               0     0         0   0     0
         NBJ.CBLINST.CBL11091.SILOG.DATA   VS     15     1 C       1   1     1
         NBJ.CBLINST.CBL11091.SILOG.INDEX  VS      1     1 T       1   1     1
         NBJ.DATASET.BASIC01.KSDS                 0     0         0   0     0
         NBJ.DATASET.BASIC01.KSDS.DATA     VS      1     1 T       1   1     1
         NBJ.DATASET.BASIC01.KSDS.INDEX    VS      1     1 T       1   1     1
         NBJ.DATASET.COPY.KSDS                    0     0         0   0     0
         NBJ.DATASET.KSDS                         0     0         0   0     0
         NBJ.DATASET.KSDS.DATA             VS      1     1 T       1   1     1
         NBJ.DATASET.KSDS.INDEX            VS      1     1 T       1   1     1
Line 1 of 61 | Col 1 of 361 | Views 1 | select * sort Entry,VSeq,Vol
```

Figure 17. Dataset List window displaying all Cluster and AIX entries beginning 'CBL.'

Panel Input Fields

`Entry>`

Specify the fileid mask.

The fileid mask represents a DSN mask that supports the following wild cards:

`*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.

`**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.

`%` A single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                    becomes:   DEV*
DEV.OEM.TRSPAN*.         becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.           becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                      becomes:   DEV.**
DEV*                     becomes:   DEV*.**
DEV.OEM.TRSPAN*          becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*              becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN           becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%              becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19               becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

`Catalog>`

Nominate a specific catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> fields.

Default is the master catalog.

Types>

Specify the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

AllVols>

Specify "Y" or "N" to control whether repeated display of the same entry occurs for multi-volume data sets. If "N" is specified, then only the primary volume entry is displayed.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	Open the File Search window for the entry.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	

	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
Entry	Char	CAT - Catalog Entry Name (usually a DSN)
Org	Enum	DS - Data Set Organiz'n PS|PO|DA|VS|etc
Trks	UInt	Alc - Total Tracks
Pri	UInt	Alc - Primary units
Alu	Char	Alc - Allocation units: C|T|B=Cyl|Trk|Blk
Sec	UInt	Alc - Secondary units
Nxt	UInt	Alc - Number of extents
Alt	UInt	Alc - Total units
Blksz	UInt	DS - Block Size
Lrecl	UInt	DS - Logical record length
RecFm	Enum	DS - Record format
PDSE	BitFlag	SMS - Partitioned dataset Extended Y|N
DsnPcu	UInt	Alc - Percent of allocated space used
DsKb	UInt	Alc - Data space used in Kilobytes
VSeq	UInt	Vol - Sequence number (Cat) 1 = First
Vol	Char	Vol - Volid
Referenced	VTOCDate	Date - Last Referenced
Created	VTOCDate	Date - Created
Expires	VTOCDate	Date - Of Expiry
BlkTrk	UInt	Alc - Blocks per track
CKDKeyL	UInt	Alc - CKD Physical Key Length
RKP	UInt	Alc - CKD Relative Key Position
DevC	Char	Alc - Device Class DASD|TAPE|etc
UnitName	Char	Alc - Device Unit Name
UnitType	Hex	Alc - Device Unit type
FSeq	UInt	Alc - Tape File Sequence number
EType	Char	CAT - Entry Type blank=NONVSAM
T	Char	CAT - Entry Type Code
DSInd	Hex	DS - Dataset flags (DS1DSIND) (hex)
RACF	BitFlag	DS - Dataset is RACF defined Y|N
DSType	Char	DS - Dataset type PDSE|KSDS|ESDS|RRDS|etc
LastVol	BitFlag	DS - Last vol holding data for dset Y|N
TBal	UInt	DS - TrackBalance: Bytes free on last trk
DCOB	BitFlag	SMS - DASDM CREATE originated blksize Y|N
DataClas	Char	SMS - Data Class
XATTR	BitFlag	SMS - Extended attributes exist Y|N

XFD	BitFlag	SMS - Extended format dataset Y|N
HFS	BitFlag	SMS - HierarchicalFileSystem Y|N
MgmtClas	Char	SMS - Management Class
ReBlk	BitFlag	SMS - May be reblocked Y|N
SMi	Hex	SMS - SMS indicators (DS1SMSFG) (hex)
StorClas	Char	SMS - Storage Class
SMS	BitFlag	SMS - System managed dataset Y|N
InICF	BitFlag	VS - Dataset cataloged in ICF catlg Y|N
IsICF	BitFlag	VS - Dataset is an ICF catalog Y|N
VS	Hex	VS - VSAM indicators (DS1OPTCD) (hex)
VTot	UInt	Vol - Total no of volumes
F1Vol	Char	Vol - Volid of 1st volume (Format 1 DSCB)
VOLSq	UInt	Vol - Sequence number (VTOC) 1 = First

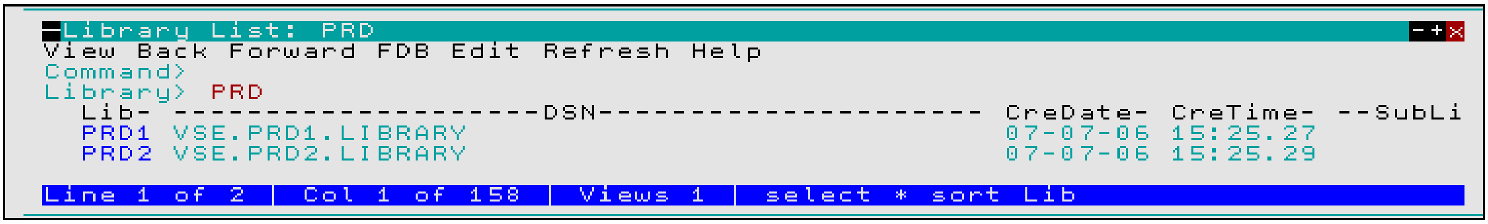
List Library Members (=3.6)

See List Windows for general features and commands common to all list windows.

The Library List window may be opened via the following:

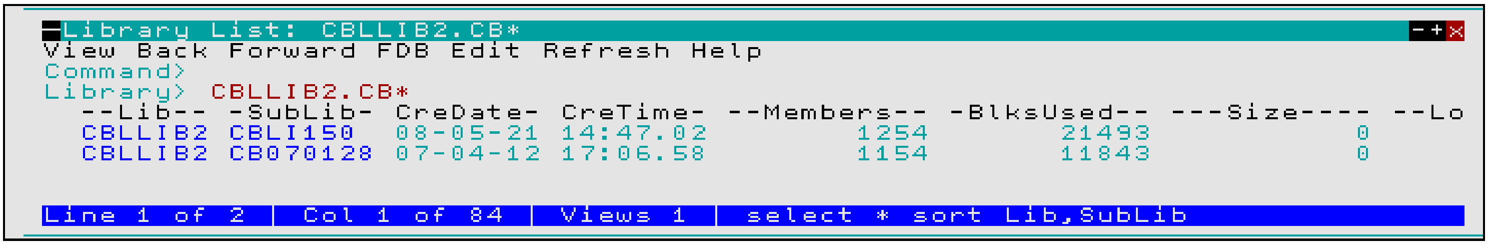
- Select option 6. 'Library' from the List Menu.
- Select 'Library Members' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LL on the command line of any window.

The Library List window displays members of a PDS/PDSE (MVS) or LIBR (VSE) library.



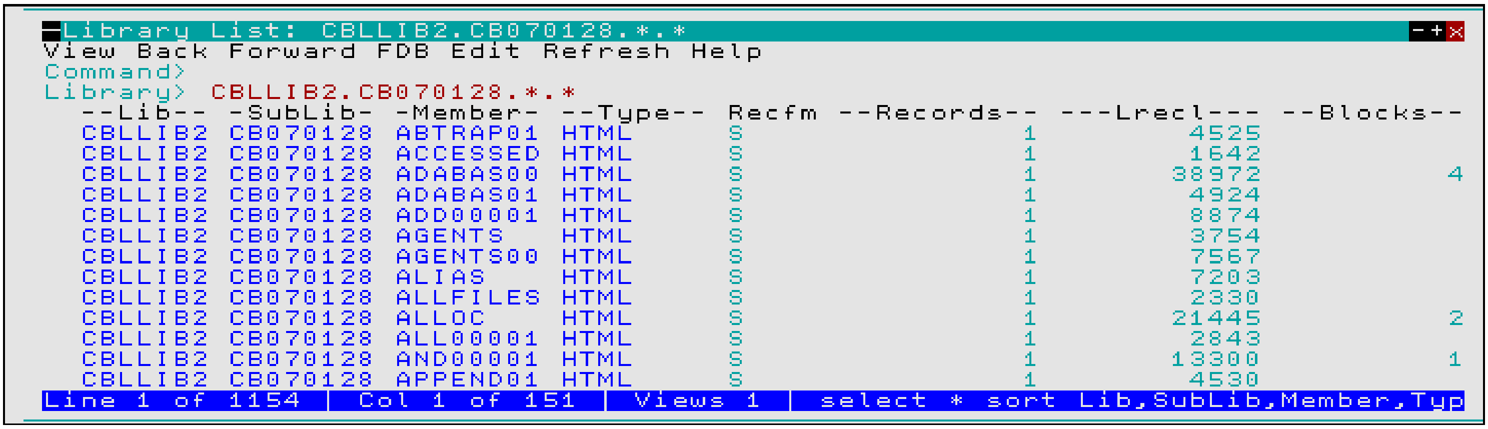
```
Library List: PRD
View Back Forward FDB Edit Refresh Help
Command>
Library> PRD
   Lib- --------------------DSN---------------------  CreDate- CreTime-  --SubLi
   PRD1 VSE.PRD1.LIBRARY                              07-07-06 15:25.27
   PRD2 VSE.PRD2.LIBRARY                              07-07-06 15:25.29

Line 1 of 2 | Col 1 of 158 | Views 1 | select * sort Lib
```

Figure 18. Library List window displaying VSE libraries beginning 'PRD'.



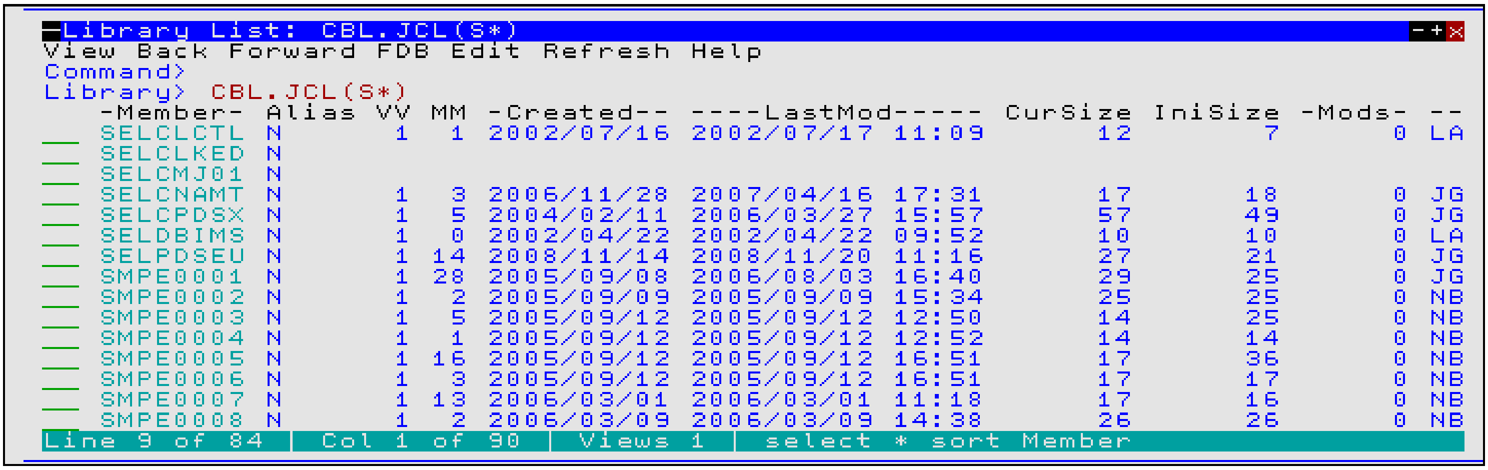
```
Library List: CBLLIB2.CB*
View Back Forward FDB Edit Refresh Help
Command>
Library> CBLLIB2.CB*
   --Lib-- -SubLib- CreDate- CreTime- --Members-- -BlksUsed-- ---Size---- --Lo
   CBLLIB2 CBLI150  08-05-21 14:47.02        1254       21493           0
   CBLLIB2 CB070128 07-04-12 17:06.58        1154       11843           0


Line 1 of 2 | Col 1 of 84 | Views 1 | select * sort Lib,SubLib
```

Figure 19. Library List window displaying VSE library 'CBLLIB2' sub-libraries beginning 'CB'.



```
 Library List:  CBLLIB2.CB070128.*.*
View Back Forward FDB Edit Refresh Help
Command>
Library> CBLLIB2.CB070128.*.*
  --Lib-- -SubLib- -Member- --Type-- Recfm --Records-- ---Lrecl--- --Blocks--
  CBLLIB2 CB070128 ABTRAP01 HTML     S               1        4525
  CBLLIB2 CB070128 ACCESSED HTML     S               1        1642
  CBLLIB2 CB070128 ADABAS00 HTML     S               1       38972          4
  CBLLIB2 CB070128 ADABAS01 HTML     S               1        4924
  CBLLIB2 CB070128 ADD00001 HTML     S               1        8874
  CBLLIB2 CB070128 AGENTS   HTML     S               1        3754
  CBLLIB2 CB070128 AGENTS00 HTML     S               1        7567
  CBLLIB2 CB070128 ALIAS    HTML     S               1        7203
  CBLLIB2 CB070128 ALLFILES HTML     S               1        2330
  CBLLIB2 CB070128 ALLOC    HTML     S               1       21445          2
  CBLLIB2 CB070128 ALL00001 HTML     S               1        2843
  CBLLIB2 CB070128 AND00001 HTML     S               1       13300          1
  CBLLIB2 CB070128 APPEND01 HTML     S               1        4530
Line 1 of 1154 | Col 1 of 151 | Views 1 | select * sort Lib,SubLib,Member,Typ
```

Figure 20. Library List window displaying all members of VSE sub-library 'CBLLIB2.CB070128'.



```
 Library List:  CBL.JCL(S*)
View Back Forward FDB Edit Refresh Help
Command>
Library> CBL.JCL(S*)
   -Member- Alias VV MM -Created-- ----LastMod----- CurSize IniSize -Mods- --
__ SELCLCTL N      1  1 2002/07/16 2002/07/17 11:09      12       7      0 LA
__ SELCLKED N
__ SELCMJ01 N
__ SELCNAMT N      1  3 2006/11/28 2007/04/16 17:31      17      18      0 JG
__ SELCPDSX N      1  5 2004/02/11 2006/03/27 15:57      57      49      0 JG
__ SELDBIMS N      1  0 2002/04/22 2002/04/22 09:52      10      10      0 LA
__ SELPDSEU N      1 14 2008/11/14 2008/11/20 11:16      27      21      0 JG
__ SMPE0001 N      1 28 2005/09/08 2006/08/03 16:40      29      25      0 JG
__ SMPE0002 N      1  2 2005/09/09 2005/09/09 15:34      25      25      0 NB
__ SMPE0003 N      1  5 2005/09/12 2005/09/12 12:50      14      25      0 NB
__ SMPE0004 N      1  1 2005/09/12 2005/09/12 12:52      14      14      0 NB
__ SMPE0005 N      1 16 2005/09/12 2005/09/12 16:51      17      36      0 NB
__ SMPE0006 N      1  3 2005/09/12 2005/09/12 16:51      17      17      0 NB
__ SMPE0007 N      1 13 2006/03/01 2006/03/01 11:18      17      16      0 NB
__ SMPE0008 N      1  2 2006/03/09 2006/03/09 14:38      26      26      0 NB
Line 9 of 84 | Col 1 of 90 | Views 1 | select * sort Member
```

Figure 21. Library List window displaying members of MVS PDSE 'CBL.JCL' whose names begin with "S".

Panel Input Fields

Library>

The name of the library for which the contents are to be listed.

◊ For **MVS** the library parameter is a PDS (or PDSE) dataset name and optionally one or more member name mask.

A member name mask supports the following wild cards:

*	A single asterisk represents an entire member name or zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS(E) DSN and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by one or more blanks and/or a "," (comma). e.g.

```
LL  DEV.OEM.CBL202.CBLI.HELP.HTML(S*AN%  WIN*, *R)
```

◊ For **VSE** the library parameter can be:

1. A library name. In this case the statistics for the library are listed. e.g.

```
LL  CBLLIB
```

2. A library name and sublibrary name. In this case the sublibrary name may be a mask containing "*" (asterisk) wild cards as supported by VSE Librarian. The statistics for all sublibraries which fit the sublibrary name mask are listed. e.g.

```
LL  CBLLIB.TEST*
```

3. A library name, sublibrary name and member name and type. In this case the member name and type may be a mask containing "*" (asterisk) wild cards. The statistics for all members which fit the mask are listed. e.g.

```
LL  CBLLIB.TEST01.*.Z
```

Prefix Line Commands

For **MVS** systems, the following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
A	Open the Create Alias dialog window.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
EX	Execute the entry. (Invokes the TSO command, EXECUTE, using the entry name as input.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FS	Open the File Search window for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

For **VSE** systems, the following prefix line commands are available when a list of VSE libraries or sublibraries is displayed:

Command	Description
<Dflt>	Prefix line command M.
M	Opens another Library List window containing the library/sub-library contents.
L	Lock the VSE LIBR member. (VSE Only)
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

For **VSE** systems, the following prefix line commands are available when a list of VSE sublibrary members is displayed:

Command	Description

<Dflt>	Prefix line command E.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
FS	Open the File Search window to search the contents of this entry. Not supported for VSE LIBR library entries.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
L	LOCK the member.
R	Rename the entry.
U	UNLOCK the member. A member may only be unlocked by the user that locked it.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

The data displayed for MVS non-LOAD libraries is:

Name	Type	Description
Member	Char	Member name
Alias	BitFlag	Alias indicator
VV	Int	Version number
MM	Int	Modification level
Created	TimeDec	Creation date
LastMod	TimeDec	Last modified date and time
CurSize	Int	Current size
IniSize	Int	Initial size
Mods	Int	Modified records
User	Char	User id

The data displayed for MVS LOAD libraries is:

Name	Type	Description
Member	Char	Member name
TTR	Hex	TTR
Rent	BitFlag	Renterable
Reus	BitFlag	Reusable
Test	BitFlag	Test module
Refr	BitFlag	Refreshable
Exec	BitFlag	Executable
Big	BitFlag	More than 16M load module
SizeHex	Hex	Contiguous storage required
EPA	Hex	Entry point address
AC	Hex	APF code
RMode	Char	Residence mode
AMode	Char	Main entry point address mode
AAmode	Char	Alias entry point address mode
AliasOf	Char	Name of aliased member

AOEPA	Hex	Entry point of aliased member
SSILvl	Hex	SSI change level
SSIFlg	Hex	SSI flag
SSISer	Hex	SSI member serial number
LMSize	Int	Large module size
LMEPA	Hex	Large module main entry point
LMAEPA	Hex	Large module alias entry point
Page	BitFlag	Page alignment required
LFmt	BitFlag	Linear format
Ovly	BitFlag	Overlay structure
Load	BitFlag	Only loadable
Scat	BitFlag	Scatter format
1Blk	BitFlag	No rld items and 1 text block
Flvl	BitFlag	Ony linkage editor F
NRLD	BitFlag	Contains no RLD items
Nrep	BitFlag	Cannot be reprocessed
TstC	BitFlag	Contains TEST cards
LnkF	BitFlag	Created by linkage editor F
Alias	BitFlag	Alias indicator

The data displayed for VSE libraries is:

Name	**Type**	**Description**
Lib	Char	Library file name
SubLib	Char	Sublibrary name
CreDate	Char	Creation date
CreTime	Char	Creation time
Members	Int	Number of members
BlksUsed	Int	Number of library blocks used
Size	Int	Sublibrary size limit
Locked	Int	Number of locked members

The data displayed for VSE sublibraries is:

Name	**Type**	**Description**
Lib	Char	Library file name
SubLib	Char	Sublibrary name
Member	Char	Member name
Type	Char	Member type
Recfm	Char	Record format
Records	Int	Number of records or bytes
Lrecl	Int	Logical record length
Blocks	Int	Number of library blocks
UpdDate	Char	Last update date
UpdTime	Char	Last update time
CreDate	Char	Creation date
CreTime	Char	Creation time
SYSIPT	BitFlag	SYSIPT data in procedure
MSHP	BitFlag	Member is MSHP controlled

MSHPByP	BitFlag	MSHP control is bypassed
PrintCC	Enum	Printer control characters
MBSTLOCK	Char	Lock identifier

List MVS Allocated Files (=3.7)

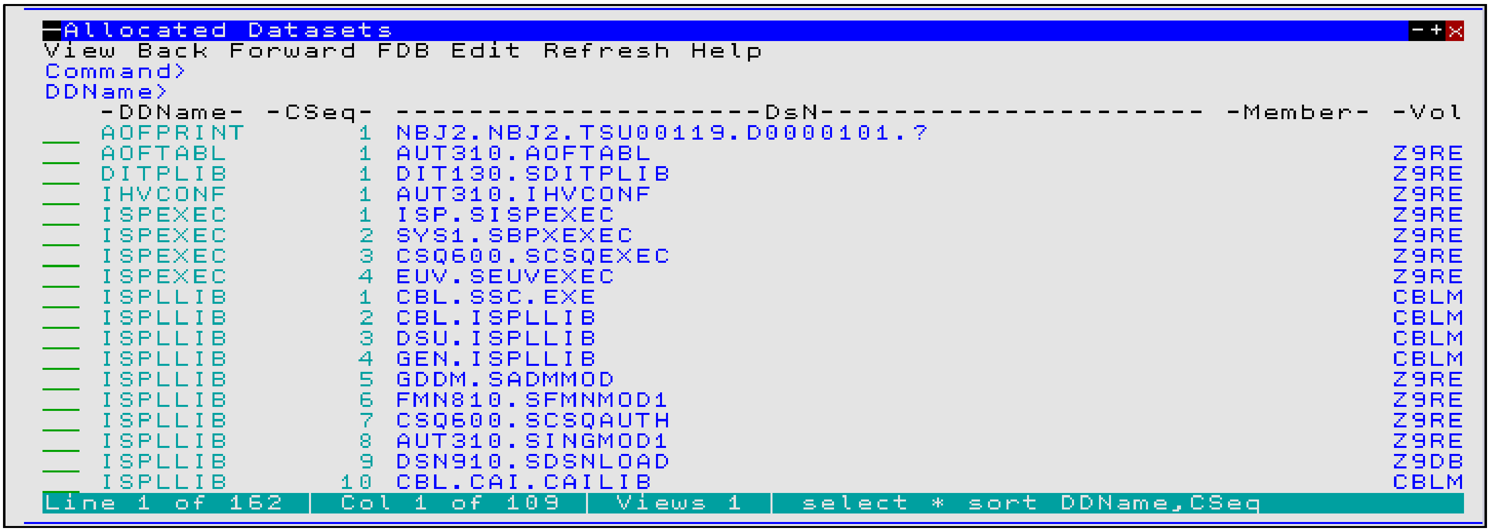
See List Windows for general features and commands common to all list windows.

The Allocated Datasets may be opened via the following:

- Select option 7. 'Allocated' from the List Menu.
- Select 'Allocated Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LA on the command line of any window.

The resultant Allocated Datasets window displays all current file allocations defined to the environment running SELCOPY/i. i.e.

MVS TSO	DDnames allocated to the TSO Userid.
MVS VTAM	Files allocated to the SELCOPY/i VTAM application.



```
 Allocated Datasets                                                                    -+x
View Back Forward FDB Edit Refresh Help
Command>
DDName>
   -DDName- -CSeq- --------------------DsN--------------------- -Member- -Vol
   AOFPRINT      1 NBJ2.NBJ2.TSU00119.D0000101.?
___ AOFTABL      1 AUT310.AOFTABL                                          Z9RE
___ DITPLIB      1 DIT130.SDITPLIB                                         Z9RE
___ IHVCONF      1 AUT310.IHVCONF                                          Z9RE
___ ISPEXEC      1 ISP.SISPEXEC                                            Z9RE
___ ISPEXEC      2 SYS1.SBPXEXEC                                           Z9RE
___ ISPEXEC      3 CSQ600.SCSQEXEC                                         Z9RE
___ ISPEXEC      4 EUV.SEUVEXEC                                            Z9RE
___ ISPLLIB      1 CBL.SSC.EXE                                             CBLM
___ ISPLLIB      2 CBL.ISPLLIB                                             CBLM
___ ISPLLIB      3 DSU.ISPLLIB                                             CBLM
___ ISPLLIB      4 GEN.ISPLLIB                                             CBLM
___ ISPLLIB      5 GDDM.SADMMOD                                            Z9RE
___ ISPLLIB      6 FMN810.SFMNMOD1                                         Z9RE
___ ISPLLIB      7 CSQ600.SCSQAUTH                                         Z9RE
___ ISPLLIB      8 AUT310.SINGMOD1                                         Z9RE
___ ISPLLIB      9 DSN910.SDSNLOAD                                         Z9DB
___ ISPLLIB     10 CBL.CAI.CAILIB                                          CBLM
Line 1 of 162 | Col 1 of 109 | Views 1 | select * sort DDName,CSeq
```

Figure 22. List MVS Allocated Datasets window.

Panel Input Fields

DDName>

Select only DDNames that match this ddname mask.

A DDName mask supports the following wild cards:

*	A single asterisk represents the complete MVS DDName or zero or more characters within the DDName mask.
%	A single percent sign represents exactly one character within the DDName mask. Up to 8 percent signs can be specified in each DDName mask.

If no wildcards are specified within the DDName mask then all MVS ddnames that **begin** with the specified DDName mask are selected.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
AS	Open an Associations list window to list associated objects for this entry.

AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS(E), open the File Search window for the entry.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
U	Unallocate the DD name.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
DDName	Char	DD name
CSeq	Int	Concatenmation sequence
DsN	Char	Dataset name
Mbr	Char	PDS member
Vol	Char	Volume serial number
Org	Char	Data set organisation
Recfm	Char	Record format
Lrecl	Int	Logical record length
BlkSize	Int	Block size
Disp	Char	Dataset disposition

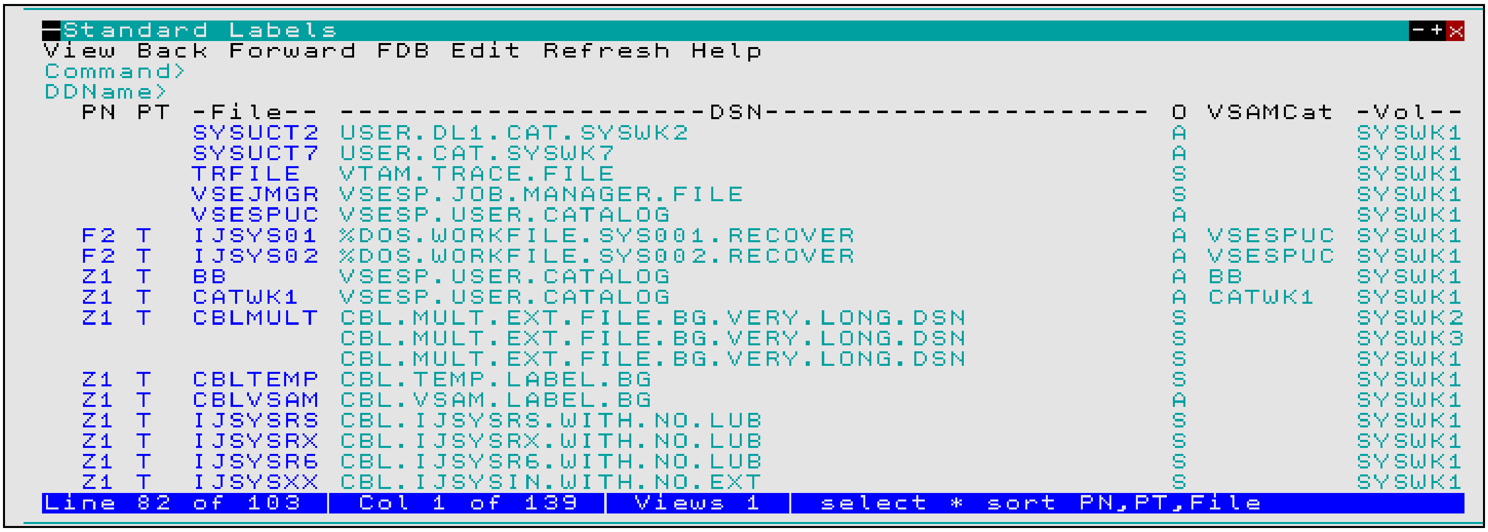
List VSE Standard Labels

See List Windows for general features and commands common to all list windows.

The VSE Standard Label window may be opened via the following to display all permanent and temporary file labels:

- Select 'Allocated Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LA on the command line of any window.

Where information for individual fields are uninitialised, then the null indicator (-1) is displayed. e.g. If EXTNO field is null, then no extents have been associated with the label.



```
 Standard Labels
View Back Forward FDB Edit Refresh Help
Command>
DDName>
  PN PT -File-- --------------------DSN--------------------- O VSAMCat -Vol--
        SYSUCT2 USER.DL1.CAT.SYSWK2                          A         SYSWK1
        SYSUCT7 USER.CAT.SYSWK7                              A         SYSWK1
        TRFILE  VTAM.TRACE.FILE                              S         SYSWK1
        VSEJMGR VSESP.JOB.MANAGER.FILE                       S         SYSWK1
        VSESPUC VSESP.USER.CATALOG                           A         SYSWK1
  F2 T  IJSYS01 %DOS.WORKFILE.SYS001.RECOVER                 A VSESPUC SYSWK1
  F2 T  IJSYS02 %DOS.WORKFILE.SYS002.RECOVER                 A VSESPUC SYSWK1
  Z1 T  BB      VSESP.USER.CATALOG                           A BB      SYSWK1
  Z1 T  CATWK1  VSESP.USER.CATALOG                           A CATWK1  SYSWK1
  Z1 T  CBLMULT CBL.MULT.EXT.FILE.BG.VERY.LONG.DSN           S         SYSWK2
                CBL.MULT.EXT.FILE.BG.VERY.LONG.DSN           S         SYSWK3
                CBL.MULT.EXT.FILE.BG.VERY.LONG.DSN           S         SYSWK1
  Z1 T  CBLTEMP CBL.TEMP.LABEL.BG                            S         SYSWK1
  Z1 T  CBLVSAM CBL.VSAM.LABEL.BG                            A         SYSWK1
  Z1 T  IJSYSRS CBL.IJSYSRS.WITH.NO.LUB                      S         SYSWK1
  Z1 T  IJSYSRX CBL.IJSYSRX.WITH.NO.LUB                      S         SYSWK1
  Z1 T  IJSYSR6 CBL.IJSYSR6.WITH.NO.LUB                      S         SYSWK1
  Z1 T  IJSYSXX CBL.IJSYSIN.WITH.NO.EXT                      S         SYSWK1
Line 82 of 103 | Col 1 of 139 | Views 1 | select * sort PN,PT,File
```

*Figure 23.*List Standard Labels Window for VSE.

Panel Input Fields

DDName>

Select only VSE label names that match this DDName mask.

A DDName mask supports the following wild cards:

*	A single asterisk represents the complete VSE label name or zero or more characters within the DDName mask.
%	A single percent sign represents exactly one character within the DDName mask. Up to 8 percent signs can be specified in each DDName mask.

If no wildcards are specified within the DDName mask then all VSE labels that **begin** with the specified DDName mask are selected.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
I	For VSAM cataloged files only, open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
VC	For VSAM cataloged files only, open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
ADisp	Char	Abend disposition
BLKSIZE	Int	SAM CKD DTFSD BLKSIZE override
BUFND	Int	VSAM ACB BUFND override. Number of Data buffers
BUFNI	Int	VSAM ACB BUFNI override. Number of Index buffers
BUFSP	Int	VSAM ACB and IDCAMS DEFINE BUFSP override
CISIZE	Int	SAM FBA DTFSD CISIZE override
DSN	Char	File dataset name
ExpDate	VTOCDate	Expiration date
ExtAlloc	Int	Number of allocated tracks/blocks
ExtNo	Int	Extent Sequence Number
ExtStart	Int	Start of extent (relative track/block number)
File	Char	File name
FBA	BitFlag	FBA Device Indicator for OPEN
LogUnit	Char	Assigned System or Programmer Logical Unit
O	Char	Open code for file type
ODisp	Char	Open disposition
PriAlloc	Int	VSAM/SAM RECORDS primary allocation
PN	Char	Partition name
PT	Char	Perm/Temp
RetPeriod	Int	Retention period in number of days. (Default 7)
RECSIZE	Int	VSAM/SAM Record size
SecAlloc	Int	VSAM/SAM RECORDS secondary allocation
TDisp	Char	Termination disposition
Vol	Char	Volume serial of this extent
VSAMCat	Char	VSAM catalog

List MVS Enqueues (=3.8)

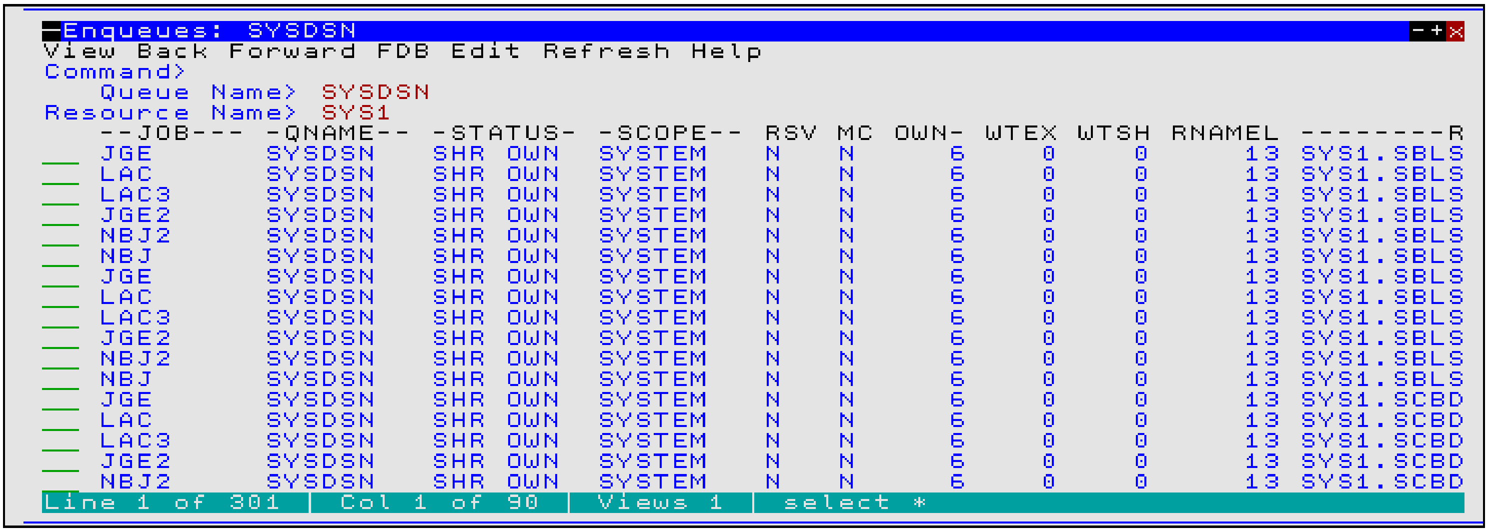
See List Windows for general features and commands common to all list windows.

The Enqueue List window may be opened via the following:

- Select option 8. 'Enqueues' from the List Menu.
- Select 'Enqueues' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LQ on the command line of any window.

The Enqueue List window displays outstanding MVS enqueues by major name and minor name (queue name and resource name).

Note: Not implemented for VSE.



```
 Enqueues: SYSDSN                                                     -+x
View Back Forward FDB Edit Refresh Help
Command>
   Queue Name> SYSDSN
Resource Name> SYS1
   --JOB--- -QNAME-- -STATUS- -SCOPE-- RSV MC OWN- WTEX WTSH RNAMEL --------R
___ JGE      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ LAC3     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ JGE2     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ NBJ2     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ NBJ      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ JGE      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ LAC3     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ JGE2     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ NBJ2     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ NBJ      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
___ JGE      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SCBD
___ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SCBD
___ LAC3     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SCBD
___ JGE2     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SCBD
___ NBJ2     SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SCBD
Line 1 of 301 | Col 1 of 90 | Views 1 | select *
```

Figure 24. Enqueue List window displaying outstanding enqueues with resource name beginning 'SYS1' in queue SYSDSN.

Panel Input Fields

`Queue Name>`
The major name (queue name) of the ENQ resource. This is a 1-8 character upper case name. For example, dataset allocations are ENQueued with resource name SYSDSN .

`Resource Name>`
This is a 1-256 character, case sensitive minor name (resource name). You need only enter the prefix of the resources you are interested in. All resources for the given queue with resource beginning with this value are listed.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
JOB	Char	Job name
QNAME	Char	Enqueue Major Name (Queue)
STATUS	Char	Status of Enqueue
SCOPE	Char	Scope of Enqueue
RSV	Char	Reserve
MC	Char	Must complete
OWN	Int	Number of owners
WTEX	Int	Number of waiters exclusive
WTSH	Int	Number of waiters shared
RNAMEL	Int	Rname length
RNAME	VChar	Enqueue Minor Name (Resource)

List MVS Job Enqueues (=3.9)

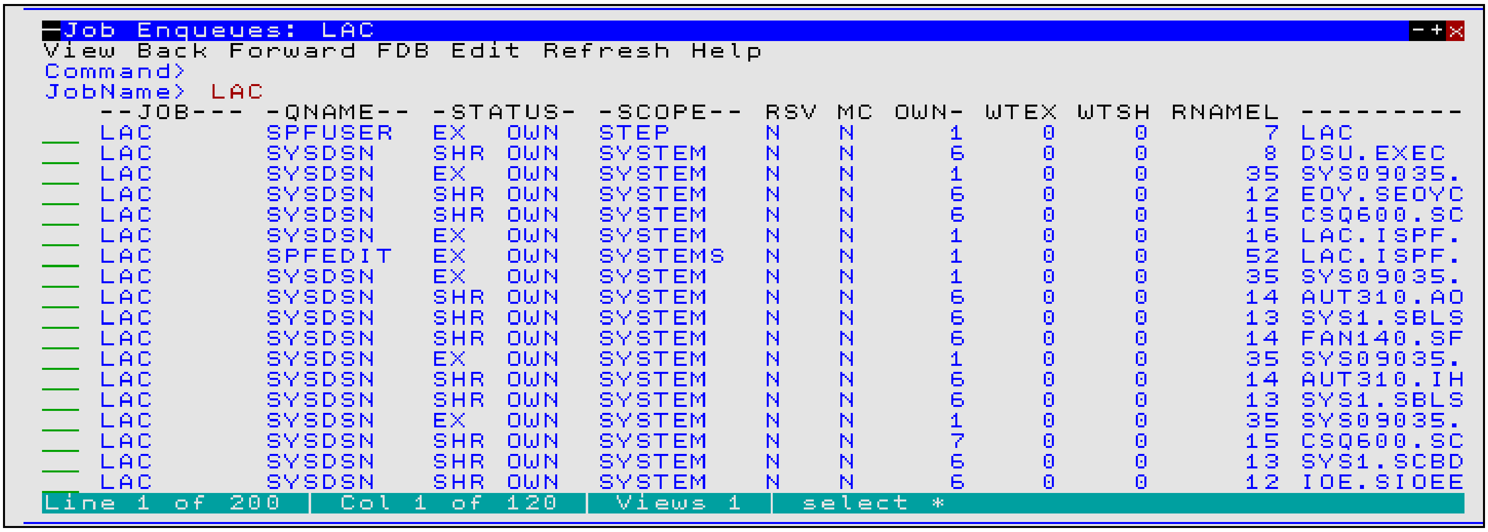
See List Windows for general features and commands common to all list windows.

The Job Enqueue List window may be opened via the following:

- Select option 9. 'Job Enqueues' from the List Menu.
- Select 'Job Enqueues' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LJQ on the command line of any window.

The Job Enqueue List window displays outstanding MVS enqueues held by a given job.

Note: Not implemented for VSE.



```
 Job Enqueues: LAC                                                        -+x
View Back Forward FDB Edit Refresh Help
Command>
JobName> LAC
   --JOB--- -QNAME-- -STATUS- -SCOPE-- RSV MC OWN- WTEX WTSH RNAMEL ---------
__ LAC      SPFUSER  EX  OWN  STEP     N   N     1    0    0      7 LAC
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0      8 DSU.EXEC
__ LAC      SYSDSN   EX  OWN  SYSTEM   N   N     1    0    0     35 SYS09035.
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     12 EOY.SEOYC
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     15 CSQ600.SC
__ LAC      SYSDSN   EX  OWN  SYSTEM   N   N     1    0    0     16 LAC.ISPF.
__ LAC      SPFEDIT  EX  OWN  SYSTEMS  N   N     1    0    0     52 LAC.ISPF.
__ LAC      SYSDSN   EX  OWN  SYSTEM   N   N     1    0    0     35 SYS09035.
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     14 AUT310.AO
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     14 FAN140.SF
__ LAC      SYSDSN   EX  OWN  SYSTEM   N   N     1    0    0     35 SYS09035.
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     14 AUT310.IH
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SBLS
__ LAC      SYSDSN   EX  OWN  SYSTEM   N   N     1    0    0     35 SYS09035.
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     7    0    0     15 CSQ600.SC
__ LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     13 SYS1.SCBD
   LAC      SYSDSN   SHR OWN  SYSTEM   N   N     6    0    0     12 IOE.SIOEE
Line 1 of 200 | Col 1 of 120 | Views 1 | select *
```

Figure 25. Job Enqueue List window displaying outstanding enqueues for Job 'LAC'.

Panel Input Fields

`JobName>`
The name of the job for which the ENQueues are to be listed.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
JOB	Char	Job name
QNAME	Char	Enqueue Major Name (Queue)
STATUS	Char	Status of Enqueue
SCOPE	Char	Scope of Enqueue
RSV	Char	Reserve
MC	Char	Must complete
OWN	Int	Number of owners

WTEX	Int	Number of waiters exclusive
WTSH	Int	Number of waiters shared
RNAMEL	Int	Rname length
RNAME	VChar	Enqueue Minor Name (Resource)

List Associations (=3.10)

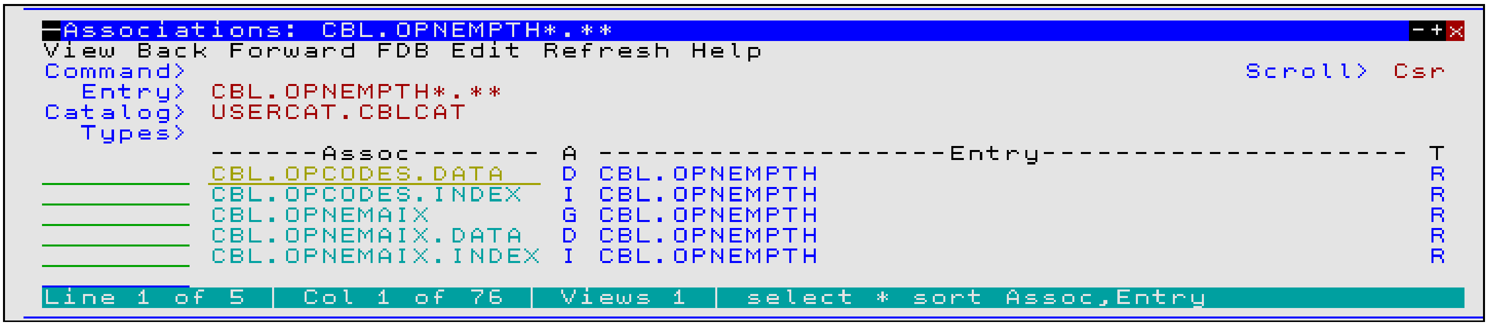
See List Windows for general features and commands common to all list windows.

The Associations List window may be opened via the following:

- Select option 10. 'Associations' from the List Menu.
- Select 'Associations' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LAS on the command line of any window.
- Enter the prefix command AS against a data set item in a list type window.

The Associations List window displays all components associated with the selected cataloged entries. e.g. A VSAM Cluster entry may display its Data, Index and any Altenate Index objects with which it is associated.

Note: Not implemented for CMS and VSE.



```
 Associations: CBL.OPNEMPTH*.**                                              -+x
View Back Forward FDB Edit Refresh Help
Command>                                                          Scroll> Csr
   Entry> CBL.OPNEMPTH*.**
Catalog> USERCAT.CBLCAT
   Types>
         ------Assoc-------  A  -------------------Entry--------------------  T
_______  CBL.OPCODES.DATA    D  CBL.OPNEMPTH                                  R
_______  CBL.OPCODES.INDEX   I  CBL.OPNEMPTH                                  R
_______  CBL.OPNEMAIX        G  CBL.OPNEMPTH                                  R
_______  CBL.OPNEMAIX.DATA   D  CBL.OPNEMPTH                                  R
_______  CBL.OPNEMAIX.INDEX  I  CBL.OPNEMPTH                                  R

Line 1 of 5 | Col 1 of 76 | Views 1 | select * sort Assoc,Entry
```

Figure 26. Associations List Window.

Panel Input Fields

Entry>

Specify the fileid mask that includes at least one cataloged object.

The fileid mask represents a DSN mask that supports the following wild cards:

`*` A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.

`**` A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.

`%` A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                becomes:   DEV*
DEV.OEM.TRSPAN*.     becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.       becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                  becomes:   DEV.**
DEV*                 becomes:   DEV*.**
DEV.OEM.TRSPAN*      becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*          becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN       becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%          becomes:   DEV.*.*SPA%*.**
```

```
SYS1.*.Z19           becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

Catalog>

Nominate a specific catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the DSN of the last catalog searched is placed in the Catalog> field.

Types>

Specify the catalog entry types for which associations will be reported. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command AS.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.

/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	**Type**	**Description**
Assoc	Char	Associated entry name
A	Char	Associated entry type code
Entry	Char	Entry name
T	Char	Entry type code

List HFS Path (=3.11)

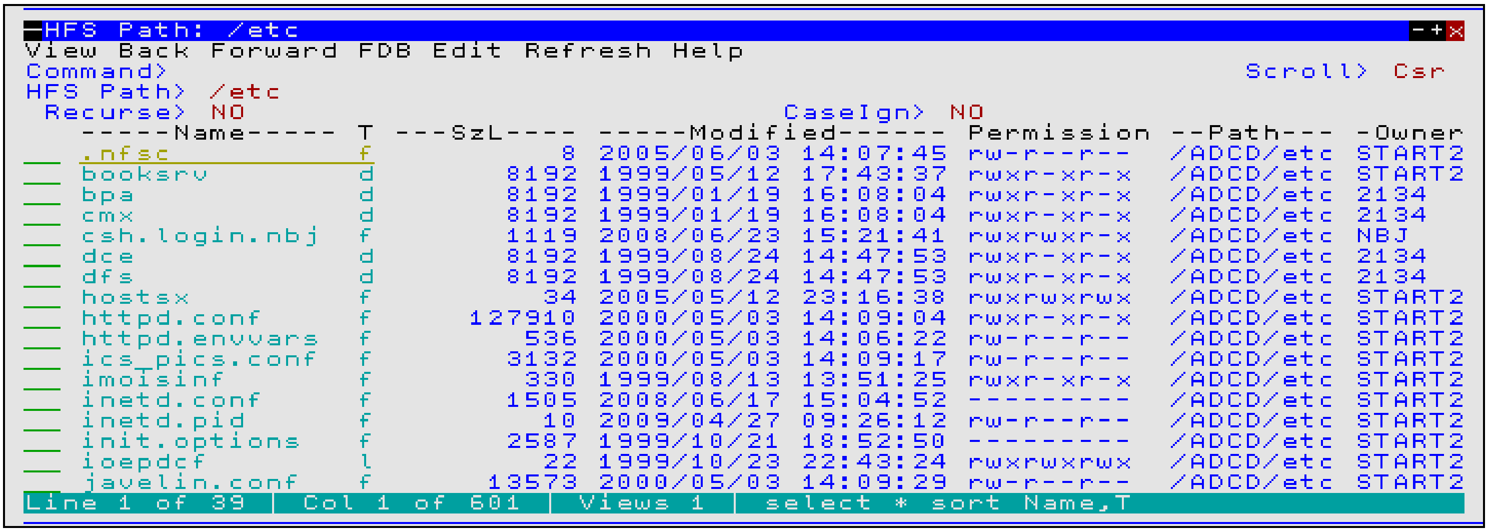
See List Windows for general features and commands common to all list windows.

The HFS Path List window displays the contents of the specified HFS directory path and optionally its sub-directories. It may be opened via the following:

- Select option 11. 'HFS' from the List Menu.
- Select 'HFS Path Details' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LP on the command line of any window.
- Enter command LD with an HFS path argument on the command line of any window.

The HFS Path List window displays file, directory and link names contained in the specified HFS path, together with stored information for each directory entry.

Note: List HFS Path is not supported for CMS or VSE.



```
 HFS Path: /etc
View Back Forward FDB Edit Refresh Help
Command>                                                          Scroll> Csr
HFS Path> /etc
 Recurse> NO                             CaseIgn> NO
   -----Name----- T ---SzL---- -----Modified------ Permission --Path--- -Owner
__ .nfsc          f          8 2005/06/03 14:07:45 rw-r--r--  /ADCD/etc START2
__ booksrv        d       8192 1999/05/12 17:43:37 rwxr-xr-x  /ADCD/etc START2
__ bpa            d       8192 1999/01/19 16:08:04 rwxr-xr-x  /ADCD/etc 2134
__ cmx            d       8192 1999/01/19 16:08:04 rwxr-xr-x  /ADCD/etc 2134
__ csh.login.nbj  f       1119 2008/06/23 15:21:41 rwxrwxr-x  /ADCD/etc NBJ
__ dce            d       8192 1999/08/24 14:47:53 rwxr-xr-x  /ADCD/etc 2134
__ dfs            d       8192 1999/08/24 14:47:53 rwxr-xr-x  /ADCD/etc 2134
__ hostsx         f         34 2005/05/12 23:16:38 rwxrwxrwx  /ADCD/etc START2
__ httpd.conf     f     127910 2000/05/03 14:09:04 rwxr-xr-x  /ADCD/etc START2
__ httpd.envvars  f        536 2000/05/03 14:06:22 rw-r--r--  /ADCD/etc START2
__ ics_pics.conf  f       3132 2000/05/03 14:09:17 rw-r--r--  /ADCD/etc START2
__ imoisinf       f        330 1999/08/13 13:51:25 rwxr-xr-x  /ADCD/etc START2
__ inetd.conf     f       1505 2008/06/17 15:04:52 ---------- /ADCD/etc START2
__ inetd.pid      f         10 2009/04/27 09:26:12 rw-r--r--  /ADCD/etc START2
__ init.options   f       2587 1999/10/21 18:52:50 ---------- /ADCD/etc START2
__ ioepdcf        l         22 1999/10/23 22:43:24 rwxrwxrwx  /ADCD/etc START2
__ javelin.conf   f      13573 2000/05/03 14:09:29 rw-r--r--  /ADCD/etc START2
Line 1 of 39 | Col 1 of 601 | Views 1 | select * sort Name,T
```

Figure 27. HFS Path List window.

Panel Input Fields

`HFS Path>`

Specify the absolute or relative HFS path name.

The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.
The following wild cards may only be specified within the name portion of the HFS path.

`*`	A single asterisk represents zero or more characters.
`%`	A single percent sign represents a single character.

`Recurse>`

Enter "YES" to recursively list the contents of all sub-directories found within the HFS path specification.
Default is "NO".

`CaseIgn>`

Enter "YES" to bypass case sensitivity for the name portion of the specified HFS path.
Default is "NO".

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Default action depends on the list entry as follows: • For a directory entry or a symbolic link to a directory, open a new List HFS list window to display the the contents of the directory. • For all other entries, a CBLe text editor view is opened to edit the data. (Equivalent to prefix command "E").
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
D	Delete the entry (file, link or directory). User will be prompted to verify the deletion.
E	Open a CBLe text editor view to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the **FSU - File Search/Update Window** to perform an advanced search and optionally update the contents of the entry.
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
SD	Open the **SDE BROWSE/EDIT Dialog Window** to browse or edit the entry's data within a Structured Data Environment **window view**.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	Type	Description
Name	ALPair	Filename
T	Enum	Dir entry type
Size	UInt	Bytes Used
Modified	Date	Data Modified Timestamp
Permission	Char	Permissions as displayed by the LS shell command.
Path	ALPair	Path
Owner	Char	Owner name
Group	Char	Group name
Fmt	Enum	File Format
Links	Int	Number of links
Mode	Int	HFS Mode (nnn)
UidX	BitFlag	Set user ID on execution

GrpX	BitFlag	Set group ID on execution
Sticky	BitFlag	Sticky Bit
INode	Hex	File Serial Number (INode)
Dev	Hex	Device ID
DevMaj	Hex	Major Device number
DevMin	Hex	Minor Device number
Uid	Int	Owner ID
Gid	Int	Group ID
Changed	Date	File Stat Chg Timestamp
Accessed	Date	Last Accessed Timestamp
Created	Date	File Creation Timestamp
BlkSz	UInt	File Block Size
AuditId	Char	RACF File ID for auditing
AA1	Hex	Auditor audit byte 1
AA2	Hex	Auditor audit byte 2
AA3	Hex	Auditor audit byte 3
AA4	Hex	Auditor audit byte 4
UA1	Hex	User audit byte 1
UA2	Hex	User audit byte 2
UA3	Hex	User audit byte 3
UA4	Hex	User audit byte 4
rU	BitFlag	Read permission for User(Owner)
wU	BitFlag	Write permission for User(Owner)
xU	BitFlag	Exec permission for User(Owner)
rG	BitFlag	Read permission for Group
wG	BitFlag	Write permission for Group
xG	BitFlag	Exec permission for Group
rO	BitFlag	Read permission for Others
wO	BitFlag	Write permission for Others
xO	BitFlag	Exec permission for Others
NoDel	BitFlag	Files should not be deleted
ShrLib	BitFlag	Shared Library
NoShrs	BitFlag	No shareas flag
Auth	BitFlag	APF authorized flag
PgmC	BitFlag	Program controlled flag
ExtLink	BitFlag	External Symbolic Link
NoDelM	BitFlag	(Mask) Files should not be deleted
ShrLibM	BitFlag	(Mask) Shared Library
NoShrsM	BitFlag	(Mask) No shareas flag
AuthM	BitFlag	(Mask) APF authorized flag
PgmCM	BitFlag	(Mask) Program controlled flag
ExtLinkM	BitFlag	(Mask) External Symbolic Link
AclAccess	BitFlag	Access ACL exists
AclFModel	BitFlag	File Model ACL exists
AclDModel	BitFlag	Directory Model ACL exists
Set	Hex	Flag bytes 1-4
FTag	Char	File Tag

BlksH	UInt	Blocks Allocated (High Order)
BlksL	UInt	Blocks Allocated (Low Order)
Opq	Hex	Opaque attribute flags
OpqM	Hex	(Mask) Opaque attribute flags
M1	Hex	HFS Mode byte 1
M2	Hex	HFS Mode byte 2
M3	Hex	HFS Mode byte 3
RefT	UInt	Reference Time
Id	Hex	File Identifier
CTime	UInt	Ctime Micro_Seconds
SecLabel	Char	Security Label
Res	Char	Reserved
Res1	UInt	Reserved
Res2	Char	Reserved
Res3	Char	Reserved

File Copy (=5)

Overview

File Copy (FCOPY) is an advanced copy utility supporting copy and optional remap of records between 2 files of potentially different data set organisations and geometry (RECFM, LRECL, BLKSIZE).

Features include:

- Use of structures to remap fields in source records to fields of the same name in destination records. Structures may be specified as an SDE structure, COBOL or PL1 copybook or a COBOL or PL1 ADATA file.
- Specification of a start record and/or a number of records to be copied so defining a subset of records to be copied/remapped.
- Append to or overwrite records in an existing target data set.
- Choose a pad character to be used to pad short records that are copied to longer fixed format records (e.g. copying an ESDS to RECFM=F; RECFM=V to RRDS or RECFM=F LRECL=80 to RECFM=F LRECL=100). Default pad character is blank (X'40).

File Copy supports copy of multiple PDS/PDSE library members to another new or existing library (Library Copy). This type of copy/remap is performed if the source file is a PDS/PDSE library, specified with or without a member mask, and the target file is a PDS/PDSE library with no member name specified. Note that a target PDS/PDSE library DSN with no member name is valid only for library copy.

File Copy:

Where the target file is not a PDS/PDSE library member, file copy supports copy of multiple PDS/PDSE library members to a single target file.

File Copy invokes the File Search/Update/Copy/Remap Utility (FSU) to copy or remap records and, on completion, returns the following summary message:

```
ZZSD356I FCOPY Summary: COPY - n records of m files.  x Remap Errors.
     y I/O Errors.
```

Library Copy:

Library copy will use the IEBCOPY facility whenever possible to perform the copy operation. It will also include copy of any member name aliases, whether or not the alias name matches the supplied member name mask. Similarly, members will be copied if their alias name matches the member name mask but the member name does not. i.e. the member name group will be copied.

Where use of IEBCOPY is not possible (e.g. libraries are of different geometry or fields in member records are to be remapped), the File Search/Update/Copy/Remap utility is invoked to copy or remap the records.

On completion of a library copy, one of the following is displayed:

- If IEBCOPY was used to perform the copy, the following informational message is returned:

  ```
  ZZSD344I FCOPY: (IEBCOPY) IGW01550I n OF m SPECIFIED MEMBERS WERE COPIED
  ```

  Furthermore, the Execute IEBCOPY window is displayed by default to report all IEBCOPY messages generated by the operation.

- If the SELCOPY/i File Search/Update/Copy/Remap utility, then, in addition to the ZZSD356I FCOPY Summary message, the following informational message is returned:

  ```
  ZZSD333I FCOPY: Members Copied=w, Replaced=x, Not Copied=y, Errors=z.
  ```

  Furthermore, the FSU - PDS Copy Statistics list window is displayed by default to report members copied and truncation/remap status of each member's records.

Source and Target File Types

The File Copy utility can copy records between any of the following file types in a single execution:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.

File Copy Panel

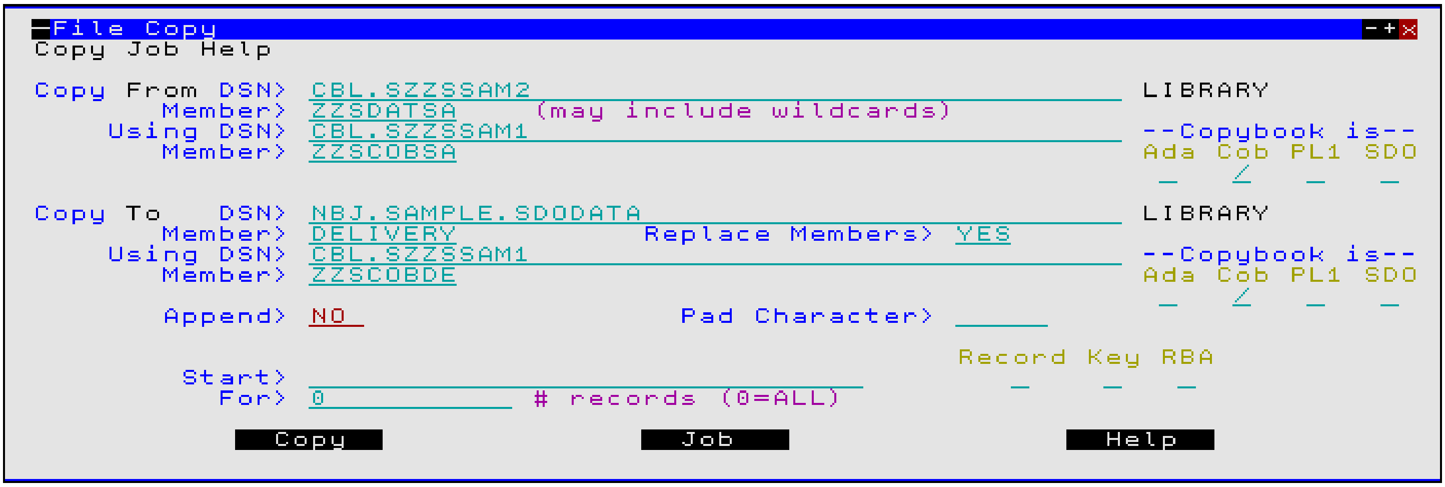
File Copy

The **File Copy** dialog window is displayed when the File Copy utility is started interactively.

This panel allows the user to invoke the File Copy utility to copy, and optionally remap, records between 2 files of potentially different data set organisations and geometry.
See *"File Copy Utility"* for an overview of functionality.

The File Copy utility dialog window may be started via the following:

- Select option 5. 'Copy/Reformat' in the SELCOPY/i Primary option menu or select option 7. 'Copy' in the Create New Datasets Menu panel.
- Select 'File Copy' from the Utilities menu.
- Execute the command FCOPY from the command line of any window.
- Execute the prefix command "**C**" from a file List type window. The resulting File Copy panel window will treat the corresponding list entry as the "From DSN" field entry.



```
-File Copy                                                          -+x
 Copy Job Help

 Copy From DSN> CBL.SZZSSAM2                                LIBRARY
         Member> ZZSDATSA     (may include wildcards)
      Using DSN> CBL.SZZSSAM1                               --Copybook is--
         Member> ZZSCOBSA                                   Ada Cob PL1 SDO
                                                             _   /   _   _

 Copy To    DSN> NBJ.SAMPLE.SDODATA                         LIBRARY
         Member> DELIVERY         Replace Members> YES
      Using DSN> CBL.SZZSSAM1                               --Copybook is--
         Member> ZZSCOBDE                                   Ada Cob PL1 SDO
                                                             _   /   _   _
         Append> NO                    Pad Character> ____

                                                 Record Key RBA
          Start>                                    _     _   _
            For> 0             # records (0=ALL)

           Copy                 Job                    Help
```

Figure 28. File Copy Dialog Window.

By default, field entries are populated with arguments and options that were entered the last time the utility panels were used.

Dialog option fields may be selected or de-selected by entering a non-blank or blank character respectively.

Unless already positioned on one of the window action buttons (Copy, Job, or Help), pressing <Enter> will first position the cursor on the "Copy" button. Pressing <Enter> with the cursor positioned on one of the action buttons will select that button to perform the relevant action. If configured, double-clicking the left mouse button on a window action button will also select that button.

As an alternative to selecting the window action buttons, the user may select an item from the menu bar.

Menu Items/Action Buttons

On selecting "Copy" or "JCL" action buttons or menu items, a check is made to determine whether the target and source files exist, and if a PDS/PDSE library, GDG or sequential data set, that the files are cataloged. Note that the File Copy utility does not currently support uncataloged data sets, however, the File Search/Update/Copy/Remap utility may be used to copy uncataloged data sets.

- If the **source** file does not exist, the user is prompted to re-enter a different fileid.

- If the **target** file does not exist and is not an HFS file, the user is prompted to identify the DSORG of the new data set (NONVSAM, KSDS, ESDS or RRDS) before being presented with the Allocate NonVSAM or Define VSAM KSDS/ESDS/RRDS/LDS dialog window as appropriate.

  If the selected DSORG is equivalent to that of the source file, then the new data set dialog window will contain values modelled on the source file. Even where the DSORG is different, dialog Record Length fields will contain an appropriate value determined from the source file.

`Copy`

Run the utility in the foreground using the current field values.

`Job`

Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the FCOPY command generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

`Help`

Display help for the File Copy Utility.

Panel Input Fields

`From DSN>`

Identifies an existing, cataloged PDS/PDSE, GDG or sequential data set; VSAM data set or HFS file to be treated as the source from which records are to be copied.
The DSORG of the specified file is automatically displayed to the right of the DSN.

`Member>`

If the DSORG of the source file is LIBRARY, this field is inspected to determine the library member mask(s) from which source members are identified.

Multiple blank and/or comma separated member masks may be specified. Each member mask may contain wildcard characters, "*" (representing zero or more characters) and/or "%" (representing exactly one character.)

If no member mask is specified, then a member mask of (*) is assumed. i.e. all members.

`Using DSN>`

If record field remapping is to be performed, this field must reference the DSN of an existing sequential data set or PDS/PDSE library containing a structure definition to be used to map data in the source records.

The structure may be a SELCOPY/i SDE structure (SDO); a COBOL or PL1 copybook or a COBOL or PL1 ADATA file.

A target file structure must also be specified for copy remap processing.

`(USING) Member>`

If the source file specified in Using DSN is a PDS/PDSE library, then this field contains the name of the library member defining the record structure.

`Ada | Cob | PL1 | SDO`

Identifies the format of the source structure file.

Ada	An ADATA file created by a previous COBOL or PL1 compilation of a copybook/include file. SELCOPY/i will use the ADATA file to generate a temporary SDO.
Cob	A COBOL copy book. SELCOPY/i will use the COBOL compiler to compile the file in order to generate a temporary SDO.
SDO	A SELCOPY/i Structured Data Object. This is the format required by SELCOPY/i to format structured data.
PL1	A PL1 include file. SELCOPY/i will use the PL1 compiler to compile the file in order to generate a temporary SDO.

These option fields correspond to SDE CREATE STRUCTURE copybook definition parameters.

`To DSN>`

Identifies the target fileid of a new or existing file, or the DSN of a new or existing PDS/PDSE library (for Library Copy), to which records or library members are to be copied.
If the file exists, the DSORG of is displayed on the right of the DSN otherwise question marks ('?') are displayed until the data set is allocated.

`Member>`

If the DSORG of the target file is LIBRARY, this field is inspected to determine the name of the library member to which records will be copied.

If no member name is specified and the source file is also a PDS/PDSE library, then Library Copy will be performed. Otherwise the target library member will be the single library member selected by the source member mask.

If no member name is specified and either the source file member mask identifies multiple members or the source file is not a PDS/PDSE library, then no error is returned but also no records are copied.

Replace Members>

Valid entries are YES or NO.
If the target file is a library (with or without an accompanying member name), then YES in this field indicates that existing members may be overwritten, NO suppresses member overwrite.

Using DSN>

If record field remapping is to be performed, this field must reference the DSN of an existing sequential data set or PDS/PDSE library containing a structure definition to be used to map data in the target file records.

The structure may be a SELCOPY/i SDE structure (SDO); a COBOL or PL1 copybook or a COBOL or PL1 ADATA file.

A source file structure must also be specified for copy remap processing.

(USING) Member>

If the target file specified in the target "Using" data set is a PDS/PDSE library, then this field contains the name of the library member defining the record structure.

Ada | Cob | PL1 | SDO

Identifies the format of the target structure file. (See source USING DSN for descriptions).

Append>

Valid entries are YES or NO.
Applicable only to file copy or remap where the target file is not a library member, YES indicates that records are to be appended to any records that may already exist in the target file. NO indicates that any existing records are to be overwritten.

If this option is not selected, then the existing records will be overwritten.

Beware that, if the DSORG of the target file is a reuseable VSAM data set (IDCAMS DEFINE REUSE), then selecting NO will overwrite **all** existing records. An attempt to overwrite an existing record will fail if the VSAM data set is defined with NOREUSE.

Selecting YES will append records to the end of the target file unless it is a VSAM KSDS file. Copy to a VSAM KSDS file will write records to their correct key positions within the data set. If a record is not in key sequence or contains a duplicate key, then that record will not be copied and the copy operation continues at the next source file record.

PAD Character>

The PAD character is used in the followying situations:

1. When a source file record is copied to a fixed length target file that has a defined length which is greater than the length of the source file record. The source record will be padded to the length of the target record using the specified pad character.

2. When a fixed length source file record is copied to a variable length target file. Trailing characters, identified by this pad character, will be stripped from the fixed length record before being copied to the target file.

The pad character may be specified as a quoted character string (e.g. 'z') or as a hexadecimal string (e.g. x'00').
Default is blank (X'40').

Start>

Identifies the record within the source file or source library members at which copying is to begin. All records prior to this record are not copied.

If no value is entered, records are copied from the first record of the source file.

Record

Select Record to indicate that the entry in the Start field is record number. The Start field entry must be specified as a numeric integer or numeric hexadecimal value (e.g. 156 or X'9C') and is suitable for any source file type.

Key

Select Key to indicate that the entry in the Start field is a VSAM KSDS or VRDS key value. The Start field entry may be specified as a character or hexadecimal string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing will occur if specified as a character string without the "C" (or "c") prefix.

The record selected will be the first record with key field data which is greater than or equal to this KEY value.

RBA

Select RBA to indicate that the entry in the Start field is an ESDS relative byte address. The Start field entry must be specified as a numeric integer or numeric hexadecimal value (e.g. 32768 or X'8000').

The record selected will be the first record with a relative byte address which is greater than or equal to this RBA value.

For>

Specifies the maximum number of records to be copied from the source file or from each selected source library member.

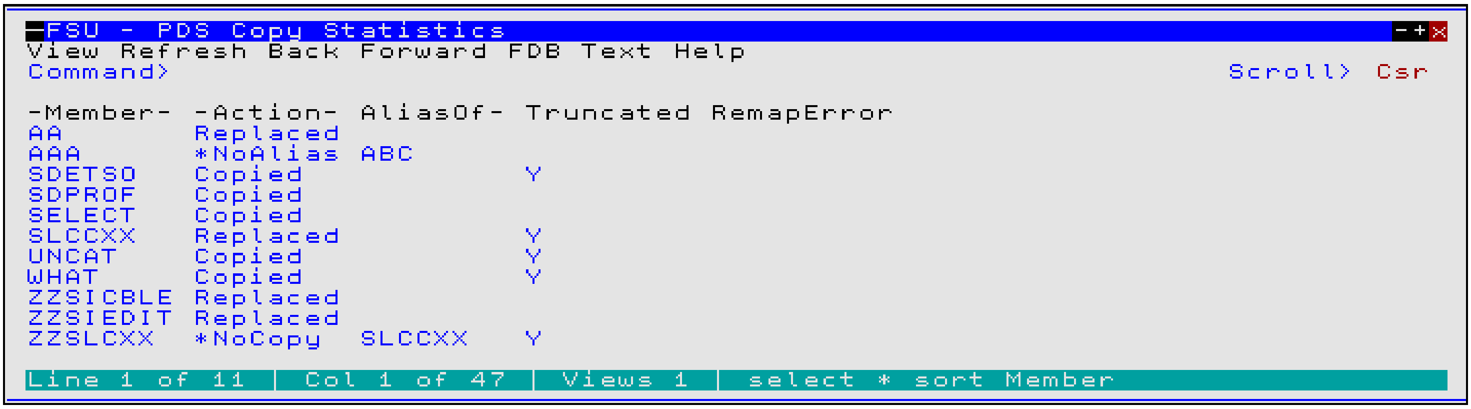
A zero (0) value indicates that no maximum limit is imposed.

File Copy - PDS Copy Statistics

Summary Format

The PDS Copy Statistics window is displayed following execution of the File Copy (FCOPY) utility if a Library Copy operation is performed that invokes the File Search/Update/Copy/Remap Utility. If the Library Copy invokes IEBCOPY, the Execute IEBCOPY window is displayed instead to report all IEBCOPY messages generated by the operation.

If FCOPY is executed in batch (using program SDEAMAIN), then the PDS copy statistics and all other messages are written to SDEPRINT when the job is submitted. If executed in the SELCOPY/i foreground, the PDS copy statistics are displayed in a list window (window class LISTFRAM).



```
=FSU - PDS Copy Statistics                                                    -+x
View Refresh Back Forward FDB Text Help
Command>                                                            Scroll> Csr

-Member- -Action- AliasOf- Truncated RemapError
AA       Replaced
AAA      *NoAlias ABC
SDETSO   Copied            Y
SDPROF   Copied
SELECT   Copied
SLCCXX   Replaced          Y
UNCAT    Copied            Y
WHAT     Copied            Y
ZZSICBLE Replaced
ZZSIEDIT Replaced
ZZSLCXX  *NoCopy  SLCCXX   Y

Line 1 of 11 | Col 1 of 47 | Views 1 | select * sort Member
```

Figure 29. PDS Copy Statistics.

Copy Statistics Fields

`Member`

Library member or alias name from the source library that has been selected for copy to the target library.

`Action`

Displays the action taken during the copy of the member or alias. Possible actions are as follow:

Copied

Member or alias was successfully copied.

Replaced

Applicable only if the REPLACE option has been specified, indicates that the copy of a member has successfully replaced a member or alias of the same name in the target library. If the source library member is an alias, then this action will occur only if it replaces an alias in the target library belonging to the same member group.

***NoRepl**

Applicable only if the REPLACE option has **not** been specified, this action indicates that the member or alias name entry cannot be copied as a member or alias of the same name already exists in the target library.

***NoCopy**

Indicates that one of the following has occurred:

1. The REPLACE option has **not** been specified and, although this member or alias name does not exist in the target library, it has failed to copy since at least one entry belonging to the same member group already exists in the target library (Action *NoRepl).
2. The REPLACE option has been specified, however at least one entry belonging to the this entry's member group exists in the target library as an entry of another member group.

***NoAlias**

Applicable only to alias entries where the REPLACE option has been specified, this action indicates that the copy failed because a member (not an alias) of the same name exists in the target library.

***StowErr**

Copy of the member or alias entry has failed due to a STOW error in attempting to write to the target PDS/PDSE library's directory. This may occur if writing a directory entry requires use of a new PDS directory block and none are available. If this is the case, a PDS library compress of the target library may resolve this problem.

`AliasOf`

If the entry is an alias, then this field displays the member name for which it is an alias.
This field is blank for non-alias library members.

`Truncated`

Contains "Y" if truncation of any of the records belonging to the member has occurred.
This field is blank if no truncation has occurred.

`RemapError`

For copy remap only, this column contains "Y" if a field in of any of the remapped records fails to be remapped due to attempted conversion of its data to an incompatible data type. If this occurs, processing stops since the incompatibility between the source and destination record fields will apply to all library members.
This field is blank if no remap error has occurred.

File Search/Update/Copy/Remap

Overview

File Search/Update/Copy/Remap (FSU) utility has more advanced functionality than the File Search utility which supports only a single search string on members of a single PDS(E) library.

Features of the File Search/Update/Copy/Remap utility include:

- Search and optionally update multiple HFS paths or multiple sequential, PDS/PDSE, GDG and/or VSAM data sets.
- Restrict PDS/PDSE library search and/or update to only members with names that match a member name mask.
- Search and optionally update uncataloged data sets by volume id(s).
- Specify the start record for both search and update operations.
- Restrict the number of records read for search and/or update.
- Restrict the search and/or update operation to a specific area of the file records.
- Apply a **structure** (copybook) overlay to map **input** file records and optionally restrict search/update to all or specific fields in records assigned to specific record types. This is known as a **Formatted File Search**/**Update**.
- For both **Unformatted** and **Formatted** input file records, optionally specify an output file to which **all** input records will be copied regardless of whether record data has been changed. This is known as **Unformatted**/**Formatted File Copy**.
- For **Formatted File Search**/**Update**, optionally specify an output file **and** output structure (copybook) to remap input record fields (i.e. alter field data type, re-order and/or delete fields) whether or not record data is changed. This is known as a **Formatted File Remap**.
- Update unformatted or formatted character data using different length search and update CHANGE strings.
- Control use of blank padding or blank absorption when character search and update CHANGE strings are of different length. Note that the CHANGE operation will fail if the length of the updated record is greater than the file's maximum record length.

Following File Search/Update/Copy/Remap execution, report output is generated in a structured format suitable for presentation to the user in an SDE window view. To generate this report output and in order to perform advanced record selection and field compare, functions and features provided by the structured data environment (SDE) are used. Therefore, the File Search/Update/Copy/Remap utility is only available to users who have a licensed version of SELCOPY installed and operational on their system.

During execution, a progress window is displayed which allows the user to interrupt processing at any point using the Attention key.

Source File Types

The File Search/Update/Copy/Remap utility can process records from any of the following file types in a single execution:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.
- (DB2 Tables planned but not yet supported).

Output Report

The report generated by the File Search/Update/Copy/Remap utility is a **structured data file**. This is designed to be browsed (not printed) from within a SELCOPY/i session using a structure definition file (SDO) which is also generated automatically during execution of the search/update.

Unless a report DSN is specified, then following execution of the utility in the foreground, the report is generated in storage and automatically displayed in a SELCOPY/i SDE edit view.

For unformatted or formatted **immediate** file update (i.e. input file record data is updated), closing the in-storage generated report will prompt the user to save the report and SDO data sets. This is so a record of file updates may be kept and, if required, may be used as input to the File Update Undo utility to roll back all updates actioned during this execution.

If required, execute command FSUEND to close the report window and save the in-storage report and SDO files generated for a file search, copy, non-immediate update or remap operation.

A list of previously generated reports is displayed on selecting "Reports" from the File Search/Update/Copy/Remap menu bar. To display a report from this list either position the cursor on the required entry then press the <Enter> key or, if configured, double-click the left mouse button on the required entry. Alternatively, generated reports may be viewed using the FSUOUT *<fsu_report_fileid>* command.

See File Search/Update/Copy/Remap Output for a detailed description of the generated output report.

Unformatted File Search/Update/Copy

Unformatted file search, update or copy is the most commonly used form of the utility, acting on **text** data files containing unformatted records.

By definition, Unformatted File Search/Update/Copy operates on records without application of a structure (SDO) or COBOL/PL1 copybook to format record data. i.e. each record is treated as a single character string.

In general, Unformatted File Search/Update/Copy processing proceeds as follows:

1. Sequentially read a record from a file matching an INPUT fileid mask.
2. Check that the record falls within the range of records to be selected for processing as specified by STARTREC/STARTKEY/STARTRBA and FOR syntax. (Note that a direct read will have been performed for VSAM KSDS, ESDS or RRDS.)
3. Apply any **search** criteria on the record as specified by a WHERE expression.
4. Apply any **search** criteria on the record as specified by one or more FIND operations.
5. Apply any data changes on the record as specified by one or more CHANGE operations.
6. If **no** OUTPUT fileid is specified and the record data has been changed by a CHANGE operation, then **update** (replace) the original input file record with the changed record.
7. If an OUTPUT fileid is specified, then **copy** the record to this file regardless of whether record data satisfies search criteria or has been changed.

Processing of the current record stops and continues with the next input record if no OUTPUT fileid is specified (for **copy**) and the record's data does not satisfy specified search criteria or if the record is not within the range of records selected for processing.

Note that search and data change functionality is based on the structured data edit (SDE) FIND, WHERE and CHANGE commands which operate on individual formatted data fields. For unformatted records, the record data occupies a single data field of data type CHAR and length equal to the file's maximum record length. This field has field reference number #1 and field name "Record", either of which may be used as a parameter to WHERE, FIND and/or CHANGE.

Unformatted File Search

Unformatted file search uses WHERE and/or FIND operations to specify search criteria and so select then report only those records that satisfy all of the search criteria.

If both WHERE and FIND criteria are specified, then a record will first have to satisfy the the WHERE expression before the FIND criteria is checked.

Unformatted File Update

Unformatted file update uses CHANGE operations to change one or more occurrences of a character search string to the specified character replace string. The changed record is then written back to the input file replacing the original record read.

Since an update-in-place is performed, the length of the updated record cannot be changed. Any CHANGE operation that results in a change to the record length will flag an error against that record in the output report.

File update should not be actioned without first performing a test run (FSU parameter NOUPDATE) where no records are actually updated but an file update output report is still generated. This allows the user to correct or accept any CHANGE errors before re-running the utility to update the records.

File update will open data sets for update-in-place processing instead of simply for input. An exclusive ENQ will be set when the data set is opened, and reset when it is closed.

Optional file search criteria, as specified for Unformatted File Search, may be used to filter input records before any CHANGE operation is performed. If no search criteria is specified then CHANGE operations will apply to all input records.

Unformatted File Copy

Unformatted file copy copies **all** records within the range of records selected for processing, from all selected input files to a single output file.

If the output file is a PDS/PDSE library then only input library members will be copied, potentially replacing existing members of the same name in the output library.

Optional change operations, as specified for Unformatted File Update, may be used to change record data as it is being copied. Records are copied regardless of whether or not they satify supplied search criteria or have been changed by a CHANGE operation.

Formatted File Search/Update/Copy/Remap

Formatted file search, update and copy is more advanced than the equivalent unformatted operations and also supports additional functionality to remap record data. Formatted operations are invoked where an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file is specified to map input record data fields for use in a search, update, copy or remap operation.

Records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records and fields within records to be searched and/or changed.

If a COBOL copybook, PL1 include file or an ADATA file (generated from a COBOL or PL1 compilation) is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO), defined by the specified or generated SDO, and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

In general, Formatted File Search/Update/Copy/Remap processing proceeds as follows:

1. Sequentially read a record from a file matching an INPUT fileid mask.
2. Check that the record falls within the range of records to be selected for processing as specified by STARTREC/STARTKEY/STARTRBA and FOR syntax. (Note that a direct read will have been performed for VSAM KSDS, ESDS or RRDS.)
3. Assign a record type (RTO) to the record.
4. Check that the record is assigned the record type specified by VIEW, otherwise the default record type.
5. Apply any **search** criteria on the record as specified by a WHERE expression.
6. Apply any **search** criteria on the record as specified by one or more FIND operations. FIND search criteria are restricted to a list of record data fields specified by SELECT.
7. Apply any data changes on the record as specified by one or more CHANGE operations. CHANGE operations are restricted to a list of record data fields specified by SELECT.
8. If **no** OUTPUT fileid is specified and the record data has been changed by a CHANGE operation, then **update** (replace) the original input file record with the changed record.
9. If an OUTPUT fileid is specified, then **copy** the record to this file regardless of whether the record is assigned the record type specified by VIEW, or whether the record data satisfies search criteria or has been changed.
10. If an OUTPUT fileid and is specified with accompanying SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file, then copy the record as described in step 9. If, however, the name of an output SDO record type matches that assigned to the record, then the field structure of the output record type will be used to **remap** the input record data fields.

Processing of the current record stops and continues with the next input record if no OUTPUT fileid is specified (for **copy** or **remap**) and the record's data does not satisfy specified search criteria, the record is not within the range of records selected for processing or if the assigned record type does not match that specified by VIEW.

Formatted File Search

Formatted file search uses WHERE and/or FIND operations to specify search criteria and so select then report only those records that satisfy all of the search criteria.

SDE WHERE and FIND operations apply only to records assigned the default record type, as identified by the VIEW operation. Fields to be searched may be identified specifically by name or reference number in the WHERE expression and/or FIND command syntax, however, FIND is restricted to only those fields identified by the SELECT operation.

Formatted search criteria are sensitive to the data type and length of the formatted fields and so appropriate action is taken when testing a field. e.g. an arithmetic compare for a numeric data field.

If both WHERE and FIND criteria are specified, then a record will first have to satisfy the the WHERE expression before the FIND criteria is checked.

Formatted File Update

Formatted file update uses CHANGE operations to change one or more occurrences of a search string to the specified replace string. The changed record is then written back to the input file replacing the original record read.

Since an update-in-place is performed, the length of the updated record cannot be changed. For formatted records, any CHANGE operation applied to the expanded data that results in a change to the **unexpanded** record length will flag an error against that record in the output report.

File update should not be actioned without first performing a test run (FSU parameter NOUPDATE) where no records are actually updated but an file update output report is still generated. This allows the user to correct or accept any CHANGE errors before re-running the utility to update the records.

File update will open data sets for update-in-place processing instead of simply for input. An exclusive ENQ will be set when the data set is opened, and reset when it is closed.

SDE CHANGE operations apply only to records assigned the default record type, as identified by the VIEW operation. Fields that are eligible for change may be identified specifically by name or reference number in the CHANGE command syntax these are restricted to only those fields identified by the SELECT operation.

Formatted data CHANGE operations are sensitive to the data type and length of the formatted fields and so appropriate action is taken when changing field data. e.g. maintain the separate length field of a changed XVARCHAR or VARCHAR field.

Optional file search criteria, as specified for Formatted File Search, may be used to filter input records before any CHANGE operation is performed. If no search criteria is specified, then CHANGE operations will apply to all input records assigned the default record type.

Formatted File Copy

The only difference between formatted and unformatted file copy is in the use of an SDE structure to optionally change field data. If a copy is to be performed without performing changes to the record data, then Unformatted File Copy may be used.

Formatted file copy copies **all** records within the range of records selected for processing, from all selected input files to a single output file.

If the output file is a PDS/PDSE library then only input library members will be copied, potentially replacing existing members of the same name in the output library.

Optional change operations, as specified for Formatted File Update, may be used to change record data as it is being copied. Records are copied regardless of whether or not they are assigned the default record type, satisfy any supplied search criteria or have been changed by a CHANGE operation.

Formatted File Remap

Formatted file remap copies **all** records within the range of records selected for processing, from all selected input files to a single output file. Furthermore, it uses an output SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to potentially remap the data fields belonging to those records assigned a record type that also exists in the output SDO.

For these records only, remap involves building a new output record using the field structure defined by the matching output record type. i.e The output record will contain **all** fields defined by this record type with field data initialised to default values.

Output record fields then inherit values from input record fields of the same name performing conversion between the input and output field data types as required. This has the effect of filtering, repositioning and reformatting data from input record fields before writing them to the output file.

Where matching input and output fields are of incompatible data types or where input data is invalid in the output field or would be truncated, then the output field does **not** inherit the input field value and a remap error is flagged for that field.

If a remap error occurs, output for the current data set is stopped, an Interrupt record is written to the report and, if output is to a PDS/PDSE library, processing continues at the next input library member.

Optional change operations, as specified for Formatted File Update, may be used to change data in the input record fields before it is copied or remapped to the output record. Records are copied or remapped regardless of whether or not they are assigned the default record type, satisfy any supplied search criteria or have been changed by a CHANGE operation.

File Search/Update/Copy Panel

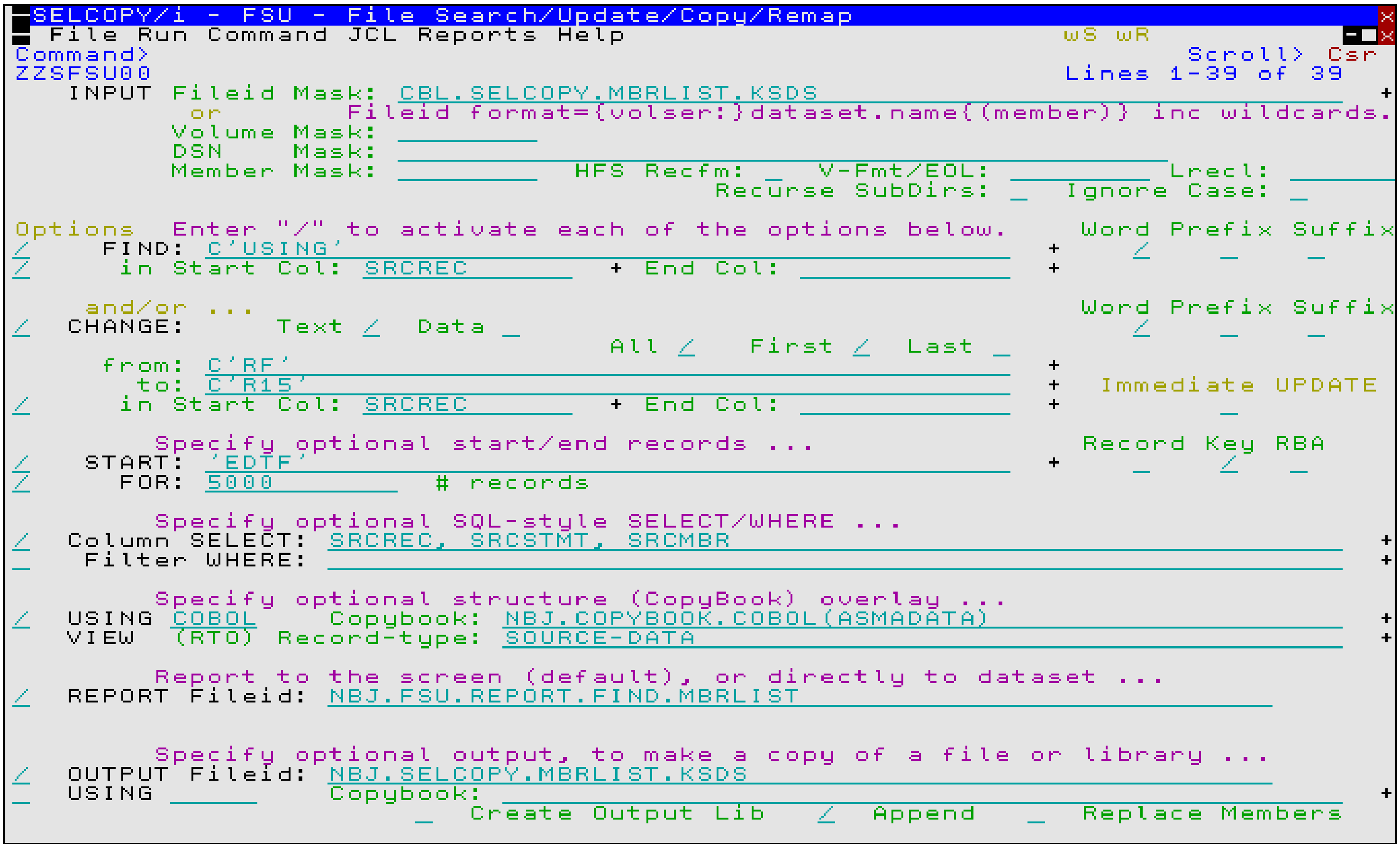
FSU - File Search/Update/Copy/Remap

The **FSU - File Search/Update/Copy/Remap** panel view (ZZSFSU00) is displayed when the File Search/Update/Copy/Remap Files utility is started interactively.

This panel allows the user to scan data sets, HFS files and/or PDS/PDSE members for search strings, optionally change data in selected records and then update the input record, or copy/remap it to an output file.
See *"File Search/Update/Copy/Remap Utility"* for a detailed description of functionality.

The File Search/Update/Copy/Remap utility panel view is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select option 6. 'Search/Update' in the SELCOPY/i Primary option menu.
- Select 'File Search/Update/Copy/Remap' from the Utilities menu.
- Execute the command FSU with no parameters from the command line of any window.
- Execute the prefix command "**F**" from a file List type window. The resulting File Search/Update/Copy/Remap panel window will treat the corresponding list entry as the INPUT fileid mask.



```
 SELCOPY/i - FSU - File Search/Update/Copy/Remap
  File Run Command JCL Reports Help                          wS wR
Command>                                                           Scroll> Csr
ZZSFSU00                                                    Lines 1-39 of 39
   INPUT Fileid Mask: CBL.SELCOPY.MBRLIST.KSDS                                +
          or       Fileid format={volser:}dataset.name{(member)} inc wildcards.
         Volume Mask:
         DSN    Mask:
         Member Mask:           HFS Recfm: _   V-Fmt/EOL:          Lrecl:
                                        Recurse SubDirs: _   Ignore Case: _

Options  Enter "/" to activate each of the options below.    Word Prefix Suffix
/    FIND: C'USING'                                        +    /    _    _
/     in Start Col: SRCREC       + End Col:                +

    and/or ...                                               Word Prefix Suffix
/  CHANGE:      Text /  Data _                                  /    _    _
                                  All /   First /  Last _
     from: C'RF'                                           +
       to: C'R15'                                          +  Immediate UPDATE
/     in Start Col: SRCREC       + End Col:                +         _

        Specify optional start/end records ...               Record Key RBA
/   START: 'EDTF'                                          +    _    /   _
/     FOR: 5000         # records

        Specify optional SQL-style SELECT/WHERE ...
/  Column SELECT: SRCREC, SRCSTMT, SRCMBR                                     +
_   Filter WHERE:                                                             +

        Specify optional structure (CopyBook) overlay ...
/  USING COBOL    Copybook: NBJ.COPYBOOK.COBOL(ASMADATA)                      +
   VIEW  (RTO) Record-type: SOURCE-DATA                                       +

        Report to the screen (default), or directly to dataset ...
/  REPORT Fileid: NBJ.FSU.REPORT.FIND.MBRLIST

        Specify optional output, to make a copy of a file or library ...
/  OUTPUT Fileid: NBJ.SELCOPY.MBRLIST.KSDS
_  USING          Copybook:                                                   +
                       _  Create Output Lib   /  Append    _  Replace Members
```

Figure 30. File Search/Update/Copy/Remap Panel.

By default, field entries are populated with arguments and options that were entered the last time the utility panels were used.

Most field entries are optional and need to be activated by entering "/" in the preceding field.

The function performed and so the type of report records generated by the File Search/Update/Copy/Remap utility is based on the fields selected. "*" (asterisk) in the following table identifies fields required to perform the specific function.

	FIND	WHERE	CHANGE	USING	OUTPUT	OUTPUT USING
Unformatted Search	*	(1)	-	-	-	-
Unformatted Search	(1)	*	-	-	-	-
Unformatted Update	(1)	(1)	*	-	-	-
Unformatted Copy	(1)	(1)	(1)	-	*	-
Formatted Search	*	(1)	-	*	-	-
Formatted Search	(1)	*	-	*	-	-
Formatted Update	(1)	(1)	*	*	-	-
Formatted Copy	(1)	(1)	(1)	(2)	*	-
Formatted Remap	(1)	(1)	(1)	*	*	*

Notes:

1. Optional field.
2. USING is required if one of the optional fields FIND, WHERE or CHANGE is specified that references formatted record fields. (If omitted, Unformatted Copy is performed.)

Note that at least one of FIND, WHERE, CHANGE or OUTPUT must be specified .

Having typed entries in the required panel fields, simply pressing the <Enter> key or, if configured, double-clicking the left mouse button will will action the utility in the foreground. During foreground execution, a progress window is displayed which allows the user to interrupt processing using the Attention key.

Alternatively, the user may select the an item from the menu bar.

Menu Bar Items

Run

Run the utility in the foreground using the current field values.

Command

Generate the FSU command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using CMDTEXT point-and-shoot execution <PF4> or copied into the user's HOME file and saved for future execution.

Command may be used when more than one FIND search criterion and/or CHANGE operation is required. The utility supports multiple FIND and CHANGE operations specified with intervening logical AND or OR operators using the FSU command syntax. This feature is not yet supported by the File Search/Update/Copy/Remap panel.

JCL

Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the FSU command generated for the specified panel field values.

Batch execution of FSU requires specification of a sequential (PS) output report data set and an implied SDO structure file having the same DSN but with suffix '.SDO'. If one or both of these data sets do not exist, then an FSURPT and/or FSUSDO DD statement is generated to allocate the specified DSN DISP=NEW.

If no report file DSN has been specified in the REPORT Fileid: field or the field has not been selected, then the user is prompted to return to the panel or accept a default DSN.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command. It also contains command FSUOUT which may be executed using CMDTEXT (<PF4>) following successful execution of the job.

Reports

Displays the **FSU List Report File panel** which displays a list of previously generated FSU report data sets. The default data set name mask used to list the report files is "%USER%.FSU.D%%%%%%.T%%%%%%" which may be changed temporarily in the **Name>** panel field.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button on the required report DSN entry will browse the report in an SDE window view.

Prefix command "K" is supported to delete (kill) a report data set and its SDO file.

Panel Input Fields

INPUT:

Identifies the data sets to be searched.

This is achieved by entering one or more fileid masks in the Fileid Mask field **or** by generating a single fileid mask using a combination of a volume mask, DSN mask and member mask in the appropriate fields.

The **Fileid Mask:**, **Volume Mask:**, **DSN Mask:** and **Member Mask:** fields all correspond to the FSU parameter INPUT.

Fileid Mask:

One or more complete fileid masks used to select the data sets and/or HFS files to be processed. Multiple fileid masks may be specified with one or more intervening blanks.

All HFS files, sequential, GDG, VSAM and PDS/PDSE data sets that match a specified fileid mask are selected for input. If one of these data sets is a PDS/PDSE library then all members of that library will be searched.

A complete fileid mask may be in one of the following formats:

1. A pre-allocated non-HFS DDNAME which may represent one or more (concatenated) data sets and/or libraries. (e.g. SYSEXEC)
2. An absolute or relative HFS Path name.

   Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in the name portion of the HFS path. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.
3. A DSN Mask and optionally a Volume and/or multiple PDS/PDSE Member Masks in the following format:

```
{volmask:}data.set.name.mask{(membmask{,membmask, ...} )}
```

Fileid masks must **not** be enclosed in quotes (TSO prefix is not used.)

In order to restrict the search to a single PDS/PDSE library and so exclude any non-PDS data set that matches the fileid mask, one or more member masks should be specified between a single pair of "( )" (parentheses). Multiple PDS/PDSE member masks must be separated by a "," (commma) and/or one or more intervening blanks.

If a volume serial mask is specified, the search is restricted to only those cataloged or uncataloged data sets that match the specified fileid mask and also have extents that exist on volume(s) that match the specified volume mask. The volume id mask is specified at the start of the fileid mask and is distinguishable from the rest of the fileid mask by an intervening ":" (colon) and no embedded blanks. e.g.

```
Z9RES1:ADCD.Z19.PROCLIB(*)
```

The fileid mask supports wild card characters as described for Volume Mask, DSN Mask and Member mask.

Examples:

```
Fileid Mask:   PE1.DEV.SRC.COBOL.CRKSW00(*)
Fileid Mask:   SYS6.JNP*.**
Fileid Mask:   OEM.TEST%%.**.CBLI.**(BOX*,D%T*,*ALL)   Z9RES1:ADCD.**
```

Volume Mask:

Optionally specify a volume serial id mask. (Not applicable to HFS files.)

The search will be restricted to only those cataloged or uncataloged data sets that match the DSN mask **and** also have extents that exist on the volume(s) that match the volume mask.

The volume mask supports wild card characters as follow:

`*` A single asterisk represents a complete volume name or zero or more characters within a volume name.
e.g. `CBL*,   *RES*`

`%` A single percent sign represents exactly one character within the volume mask.
e.g. `Z9DB9%,   %%XV3%`

DSN Mask:

The data set name or HFS fileid mask used to identify data sets to be searched.

A DSN mask is mandatory if a Fileid Mask is not specified. A pre-allocated DDNAME may be specified in the DSN mask field to represent one or more (concatenated) data set and/or library.

All HFS files, sequential, GDG, VSAM and PDS/PDSE data sets that match a specified fileid mask are selected for input. If one of these data sets is a PDS/PDSE library then all members of that library will be searched.

A DSN mask must **not** be enclosed in quotes (TSO prefix is not used) and supports wild card characters as follow:

`*` A single asterisk represents a DSN qualifier or zero or more characters within a DSN qualifier.
e.g. `DEV.CBLINS.*.JCL,   DEV.CBLINS.TEST*.ISP*LIB,   DEV.CBLINS.*.*`

`**` Double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
e.g. `DEV.CBLINS.**,   DEV.CBLINS.**.CBLE`

`%` A single percent sign represents exactly one character other than "." (dot/period) within a DSN qualifier.
e.g. `DEV.CBLINS.TEST0%.JCL,   DEV.CBLI%%.TEST06.CBLI.%%%`

Member Mask:

Optionally specify one or more PDS/PDSE member name masks.

In order to restrict the search to PDS/PDSE libraries and so exclude any non-PDS data set that matches the DSN mask, one or more member masks should be specified. Multiple PDS/PDSE member masks must be separated by a "," (comma) and/or one or more intervening blanks.
e.g. `BLOCK, PROFILE   BOXSEQ`

The search will be restricted to only those PDS/PDSE data sets that match the DSN mask **and** only members with a member name that matches any one of the supplied member masks.

A member mask supports wild card characters as follow:

`*` A single asterisk represents an entire member name or zero or more characters within a member name.
e.g. `CBL*5,   BOX*,   D*T*`

`%` A single percent sign represents exactly one character within a member name mask.
e.g. `H%,   D%R*,   E%A`

HFS

HFS specific options that apply to **all** HFS path fileid masks specified in the INPUT field.

The HFS fields are as follow:

Recfm:

Specify the record format F (Fixed) or V (Variable) to be used for all HFS files that match the HFS path fileid masks specified by the INPUT field.
If left blank, the default record format U (Undefined) is implied using EOL characters to delimit record data.

This field corresponds to the FSU HFS Option parameter RECFM.

V-Fmt/EOL:

For record format U (RecFm fileid is blank), this field specifies the EOLIN (input end-of-line) delimiter to be used for records in all HFS files that match the HFS path fileid mask(s) specified by the INPUT field.
If left blank, the default EOL=STD is implied which automatically identifies the EOL delimitter for each input HFS file.

For record format V, the contents of this field must be of the format *(offset,length,origin)* where enclosing "( )" (parentheses) are optional. *offset* and *length* define the record length field in the record data, while *origin* defines the start of the record data to which the record length is applied.
If left blank, the default values are (0,2,0) which describes standard RECFM V organisation data sets.

For record format F, the contents of this field is ignored.

This field corresponds to the FSU HFS Options parameter RECFM *rfmstr* or EOL *eolstr*.

Lrecl:

Specify the maximum record length of input records belonging to all HFS files that match the HFS path fileid masks specified by the INPUT field.

Default for *lrecl* and its effect on input records is as supported by the SDE CLI command EDIT.

This field corresponds to the FSU HFS Options parameter LRECL *lrecl*.

Ignore Case:

Bypass case sensitivity for the **name** portion of all specified HFS path fileid masks specified in the INPUT field. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.

This field corresponds to the FSU HFS Options parameter CASEIGN.

Recurse SubDirs:

For all HFS path names specified in the INPUT field, recursively search files within all sub-directories found within each HFS path specification.

This field corresponds to the FSU HFS Options parameter RECURSE.

FIND:

Insert arguments to be translated into a single SDE FIND operation.

If activated, the FIND operation defines a search criterion for Unformatted File Search or Formatted File Search, or is used for record filtering before executing a CHANGE operation.

The FIND operation is performed only on those input records that first satisfy any supplied WHERE expression and, for formatted records, match the VIEW default record type selected. For unformatted records where no WHERE expression is specified, FIND will be performed on all input records.

If no CHANGE operation is specified, then a simple file search is performed so that records that satisfy the FIND operation are reported. Otherwise, records that satisfy the FIND operation are processed by the CHANGE operation.

For Formatted records, a numeric search string will be treated as a signed numeric value and an arithmetic compare will occur for numeric data fields. For non-numeric fields and unformatted records, all search strings are treated as character data and a logical string compare is performed.

If more than one FIND search string is required, enter details of the first search string in the FIND field then select the "Command" menu bar item to edit the generated FSU command and so add further FIND conditions. Note that each group of FIND parameters that constitute a single FIND condition, must be separated by either a logical AND or a logical OR and must be enclosed in "( )" (parentheses).
e.g. `FIND (   (c'James Kirk')   AND   (c'Spock' WORD)   )`

If logical AND is specified, all FIND conditions must be satisfied before the record is selected. If logical "OR" is specified, any of the FIND conditions must be satisfied before the record is selected. FIND conditions separated by a mixture of logical "AND" and logical "OR" operators is not permitted. If this is required, the WHERE clause should be used in place of, or in addition to, the FIND.

The FIND field corresponds to the FSU parameter FIND.

The FIND fields are as follow:

FIND:

Specifies a single SDE FIND command search string.

In order to include a FIND operation (and hence any of the other fields associated with FIND), the **FIND** field must be enabled by entering "/" in the activation field.

This field corresponds to SDE FIND command search *string*.

Start Col:

Specifies the start (or only) data position or formatted record field within from which the scan for the search string will begin. Record data in positions or fields that occur before this start column value is not searched.

If a numeric value is entered, then it is treated as a position in the input records.
For formatted records, this field may contain a field name (e.g. Emp_Name) or a field reference number (e.g. #5) instead of a position. If a multiple field range or a more complex combination of field references is required, then use the "Command" menu bar item to edit the FSU CLI command and make the relevant additions.

In order to use this start position/field value and optionally an end position/field value, then the field(s) must be enabled by entering "/" in the activation field.

This field corresponds to SDE FIND parameter *pos1* or *field_col*/*field_col1*.

End Col:

Specifies the end data position or formatted record field within the input records beyond which no part of the search string will be found. Only positions or fields that are between the start and end position/field values will be searched. Record data in positions or fields following this end column value is not searched.

The format of the value entered in this field (i.e. position or field) must match that entered for the Start Col field.

If no start and end position/field is entered, the default start position is 1 and the default end position is the length of the record.

If no end position/field is entered then:

- If a start position is specified, the default end position is the start position plus the length of the search string minus 1.
- If a start field is specified, only that field will be searched.

This field corresponds to SDE FIND parameter *pos2* or *field_col2*.

Prefix | Suffix | Word

If the condition is to include one of the parameters PREFIX, SUFFIX or WORD, then enter "/" in the appropriate parameter field. Note that these fields are mutually exclusive.

PREFIX	A successful match only if the search string is found at the start of a word and does not constitute the entire word.
SUFFIX	A successful match only if the search string is found at the end of a word and does not constitute the entire word.
WORD	A successful match only occurs the search string constitutes an entire word.

If none of these fields are selected, the search string may be found anywhere within the data being searched.

These option fields correspond to SDE FIND parameters PREFIX, SUFFIX and WORD.

For a detailed description of FIND parameters and their effects, please refer to the SDE FIND command documentation.

CHANGE:

Insert arguments to be translated into a single SDE CHANGE operation.

In order to include a CHANGE operation and any of its associated fields, the **CHANGE** field must be enabled by entering "/" in the activation field.

If activated, CHANGE will perform character string or numeric value substitution on record data and, where OUTPUT fileid has **not** been specified, implies Unformatted File Update or Formatted File Update.

Note that, for File Update only, the CHANGE operation must not alter the length of an unexpanded record, otherwise a change error will occur. This condition will be flagged against the record in the output report.

The CHANGE operation is performed only on those input records that first satisfy any supplied (WHERE and/or FIND) search criteria and, for formatted records, match the selected VIEW default record type. For unformatted records, where no FIND operation or WHERE expression is specified, CHANGE will be performed on all input records.

For Formatted records, a numeric search string and replace string will be treated as signed numeric values. An arithmetic compare will occur for the search string when applied to numeric data fields and the numeric replace string converted to a field's numeric data type as appropriate. For non-numeric fields and unformatted records, all search and replace strings are treated as character data and a logical string compare is performed.

If more than one CHANGE operation is required, enter details of the first search and replace strings in the CHANGE field then select the "Command" menu bar item to edit the FSU command and so add further CHANGE conditions. Note that each group of CHANGE parameters that constitute a single CHANGE operation, must be separated by either a logical "AND" or a logical "OR" and must be enclosed in "( )" (parentheses).
e.g. `CHANGE (   ('01656' '+1656' PREFIX ALL)   OR     ('-1656' '+1656' PREFIX ALL)   )`

If logical "AND" is specified, a CHANGE operation will be executed for all of the CHANGE conditions. If logical "OR" is specified, a CHANGE operation will be executed for each CHANGE condition in turn until one is successful. CHANGE conditions separated by a mixture of logical "AND" and logical "OR" operators is not supported.

The CHANGE field corresponds to the FSU parameter CHANGE.

The CHANGE fields are as follow:

`Text | Data`

Determines the effect on data in a record or character field following a successful CHANGE operation where the replace string is of a different length to the search string.

TEXT indicates that the field/record data is formatted so that positions of words, prefixed by multiple consecutive blanks, are preserved whenever possible. Blanks are absorbed or inserted as appropriate to maintain word positions within the record.

DATA indicates that the field/record data is unformatted so that all data following the changed string is shifted left or right as appropriate.

These option fields correspond to SDE CHANGE parameters TEXT and DATA.

`All | First | Last`

Indicates whether ALL occurrences, the FIRST occurrence or the LAST occurrence of the CHANGE search string within the input record is to be changed. Since each execution of CHANGE operates on the full width of selected data within the record, NEXT and PREV CHANGE parameters are equivalent to FIRST and LAST respectively and so are not included as CHANGE options.

Unlike the CHANGE commmand which in a single invocation, is applied to all appropriate records in an SDE edit window in a single invocation, execution of a CHANGE operation via the File Search/Update/Copy/Remap utility processes one record at a time. Therefore, FIRST and LAST does **not** refer to the first and last occurrence in the file, but the first and last occurrence within the selected input record or input record fields.

CHANGE ALL is selected by default. If CHANGE FIRST or LAST is required, then enter "/" in the appropriate parameter field. Note that these fields are mutually exclusive.

These option fields correspond to SDE CHANGE parameters ALL, FIRST and LAST.

`from:`

Specifies a single SDE CHANGE command search string (*string1*).

`to:`

Specifies a single SDE CHANGE command replace string (*string2*).

`Start Col:`

Specifies the start (or only) data position or formatted record field within the from which the scan for the CHANGE search string will begin. Record data in positions or fields that occur before this start column value is not searched.

If a numeric value is entered, then it is treated as a position in the input records.
For formatted records, this field may contain a field name (e.g. Emp_Name) or a field reference number (e.g. #5) instead of a position. If a multiple field range or a more complex combination of field references is required, then use the "Command" menu bar item to edit the FSU CLI command and make the relevant additions.

In order to use this start position/field value and optionally an end position/field value, then the **Start Col** and **End Col** field(s) must be enabled by entering "/" in the activation field.

This field corresponds to SDE CHANGE parameter *pos1* or *field_col*/*field_col1*.

`End Col:`

Specifies the end data position or formatted record field within the input records beyond which no part of the CHANGE search string will be found. Only positions or fields that are between the start and end position/field values will be searched. Record data in positions or fields following this end column value is not searched.

The format of the value entered in this field (i.e. position or field) must match that entered for the Start Col field.

If a character CHANGE search string is found within a start/end positional range, then the found text will be replaced even if the replacement string extends beyond the end position of the range.

If no start and end position/field is entered, the default start position is 1 and the default end position is the length of the record.

If no end position/field is entered then:

- If a start position is specified, the default end position is the start position plus the length of the search string minus 1.
- If a start field is specified, only that field will be searched.

This field corresponds to SDE CHANGE parameter *pos2* or *field_col2*.

`Prefix | Suffix | Word`

If the condition is to include one of the parameters PREFIX, SUFFIX or WORD, then enter "/" in the appropriate parameter field. Note that these fields are mutually exclusive.

PREFIX	A successful match only if the CHANGE search string is found at the start of a word and does not constitute the entire word.
SUFFIX	A successful match only if the CHANGE search string is found at the end of a word and does not constitute the entire word.
WORD	A successful match only occurs the CHANGE search string constitutes an entire word.

If none of these fields are selected, the CHANGE search string may be found anywhere within the data being searched.

These option fields correspond to SDE CHANGE parameters PREFIX, SUFFIX and WORD.

`Immediate UPDATE`

Applicable to Unformatted or Formatted File Update only, records are not actually updated unless this field option is selected. This allows the user to perform a test update run, review the output report and make any corrections before running the utility again to perform the record updates. It is highly recommended that this should be done prior to executing the file update with Immediate UPDATE enabled.

Caution: Activation of this field will cause all changed records to be re-written.

As a precaution, the utility panel will prompt the user to confirm this selection before executing the update or generating the FSU command.

This option field corresponds to FSU parameter UPDATE/NOUPDATE.

For a detailed description of CHANGE parameters and their effects, please refer to the SDE CLI CHANGE command documentation.

`START:`

For every file matching an INPUT fileid mask, this field defines the record at which search/update/copy/remap processing will start.

In order to provide a start record, the **START** field must be enabled by entering "/" in the activation field.

User should enter a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the FSU parameters STARTREC *recno*, STARTRBA *rba* and STARTKEY *key*.
There is no default.

The START fields are as follow:

`Record | Key | RBA`

Identifies the type of start value as described by Start above. Enter "/" in the appropriate, mutually exclusive parameter field.

`FOR:`

For every file matching an INPUT fileid mask, this field specifies the number of records, beginning at the start record, on which search/update/copy/remap processing will occur.

In order to provide a number of records limit, the **FOR** field must be enabled by entering "/" in the activation field.

This field corresponds to the FSU parameter FOR.
Default is all records.

`Column SELECT:`

For formatted records only, specifies SDE SELECT command syntax to select individual fields from records assigned the default record type as specified by VIEW.

In order to include field selection, the **Column SELECT** field must be enabled by entering "/" in the activation field.

Only those fields specified by SELECT are eligible for inclusion in the FIND and CHANGE operations. i.e. If the SELECT field is enabled, all record field entries referenced by Start Col and End Col for FIND and/or CHANGE must exist in the list of selected record fields.

The order in which fields are specified by the SELECT field is the order in which fields are processed. Furthermore, "*" (asterisk) may be specified to represent all remaining fields in the default record type that have so far not been selected.

This field corresponds to the FSU parameter SELECT.
Default is the first record type defined in the structure (SDO).

Filter WHERE:

Specifies an SDE expression to be used on an SDE WHERE record filtering operation.

In order to include record filtering, the **Filter WHERE** field must be enabled by entering "/" in the activation field.

If activated, the WHERE expression defines search criteria for Unformatted File Search or Formatted File Search, or is used for record filtering before executing a FIND or CHANGE operation.

For **formatted** records, the WHERE operation is performed only on those input records that are assigned the default record type as specified by VIEW. Any field within the formatted record may be referenced regardless of whether it has been included by the SELECT.
e.g. `WHERE ( (#3 >= 22) AND ( (EmpName = 'Smith') OR (Dept >> 'E1') ) )`

For **unformatted** records, the WHERE operation is performed on all records. The WHERE expression may only include reference to a single field (field reference #1, field name "Record") which evaluates to all data in the focus record. This field has a data type of CHAR and length equal to the file's maximum record length.
e.g. `WHERE ( (#1 >> '01') AND (words(#1) = 5) )`

If no CHANGE operation is specified, then a simple file search is performed so that the report identifies all records that satisfy the WHERE operation and any subsequently executed FIND operation. If a CHANGE operation is specified, only these records are eligible to be changed and only records that have been changed are identified in the output report.

The WHERE field corresponds to the FSU parameter WHERE.

USING

Specifies the SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map input record data fields for use in a Formatted File Search, Formatted File Update, Formatted File Copy or Formatted File Remap operation.

In order to specify an input structure and so use formatted record processing, the **USING** field must be enabled by entering "/" in the activation field.

Where USING has been enabled, records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records, and fields for WHERE, FIND and CHANGE operations.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. This is an overhead which may be overcome by generating a non-temporary SELCOPY/i SDO file and referencing this SDO instead. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO) defined in the specified or generated SDO and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

This field corresponds to the FSU parameter USING SDO/COBOL/PL1/ADATA *in_struct*.
Default is not to apply a structure and so use **unformatted** processing.

The USING fields are as follow:

SDO | COBOL | PL1 | ADATA

Identifies the source format of the input structure file.

SDO	A SELCOPY/i Structured Data Object. This is the format required by SELCOPY/i to format structured data.
COBOL	A COBOL copy book. SELCOPY/i will use the COBOL compiler to compile the file in order to generate a temporary SDO.
PL1	

	A PL1 include file. SELCOPY/i will use the PL1 compiler to compile the file in order to generate a temporary SDO.
ADATA	An ADATA file created by a previous COBOL or PL1 compilation of a copybook/include file. SELCOPY/i will use the ADATA file to generate a temporary SDO.

These option fields correspond to SDE CREATE STRUCTURE copybook definition parameters.

Copybook:

Specifies the DSN and, if applicable, the member name of the input structure file.

VIEW (RTO) Record Type:

Applicable to **formatted** records only, the **VIEW (RTO) Record Type** field optionally specifies the name of a single record type (RTO) to be used on an SDE VIEW operation.

The specified RTO becomes the default record type against which all SELECT, WHERE, FIND and CHANGE operations are performed. Records not assigned this record type are not processed by these operations.

The named record type must match one defined within the specified SDO structure or within a temporary SDO structure generated from a COBOL, PL1, ADATA file referenced by USING.

In order to specify a default record type, formatted record processing must have been enabled by entering "/" in the activation field for **USING**.

The VIEW field corresponds to the FSU parameter VIEW *record_type*.

Although not mandatory, it is recommended that a VIEW entry is specified when performing formatted data processing so avoiding confusion over which records are to be processed.
Default is the first record type defined in the structure (SDO).

REPORT Fileid:

Specifies the DSN of a sequential (PS) data set to which the file search/update/copy/remap report records will be written. The dataset name must be fully qualified, quotes being unnecessary but permitted.

If executing the utility in the foreground, the specified report data set must already exist. However, if the report data set does not exist and a batch job stream is to be generated (menu item JCL), then FSURPT and FSUSDO DD statements are generated as appropriate with DISP=NEW.

The report is a structured data file designed to be browsed (not printed) using an SDE structure definition object (SDO), which will be generated automatically.

The associated SDO fileid is constructed simply by adding **.SDO** to the report fileid. Therefore, the DSN of the report file is restricted to 40 bytes in length.
Report output to an HFS dataset is not currently supported.

In order to specify a non-default output report data, the **REPORT Fileid** field must be enabled by entering "/" in the activation field.

The REPORT field corresponds to the FSU parameter REPORT *fileid*.

If this option is not specified, *fileid* defaults to "*user*.FSU.Dyyyyddd.Thhmmss" with SDO fileid "*user*.FSU.Dyyyyddd.Thhmmss.SDO".

OUTPUT Fileid:

Specifies an output file to which **all** input records will be copied. The output file may be one of the following:

◊ An existing physical sequential (PS) data set.
◊ An existing VSAM data set.
◊ A new or existing member of an existing PDS/PDSE library.
◊ A new or existing HFS/ZFS file path.
◊ If input is a PDS/PDSE library, a new or existing PDS/PDSE library.

If activated, specification of an output fileid alone defines the utility processing to be Unformatted File Copy or Formatted File Copy. If an Output USING structure is also specified, then processing will be Formatted File Remap.

If a CHANGE operation is specified and activated, then record data may be changed before it is written to the output file.

The format of the output file should be compatible with input record data. e.g.

- If an output KSDS data set is specified, input records must be in key sequence, as defined by the output file, and must not contain duplicate keys. If an input record does not satisfy these conditions, it will fail to copy.
- Records will be truncated if the input record length exceeds the maximum allowed by the output file.

In order to specify an output file and so use copy processing, the **OUTPUT Fileid** field must be enabled by entering "/" in the activation field.

If this option is not specified, then file search or file update processing is performed.

The OUTPUT Fileid field corresponds to the FSU parameter OUTPUT *fileid*.

The Output Fileid fields are as follow:

`USING`

Specifies the SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map output record data fields for use in Formatted File Remap.

Formatted File Remap processing only occurs when an output file and input and output structures are supplied. Therefore, an output structure is ignored if no input structure has been specified.

During the remap process, the following will occur:

1. Input records of record type not defined in the output structure are copied without field remap.
2. Output structure record types not defined in the input structure are redundant and so are ignored.
3. Record data in input fields are copied to output fields of the same name belonging to record types of the same name.
4. The input field data will be reformatted to the data type of the output field and will be moved to the output field's position within the record map.
5. Any input fields whose field names are not part of the output record structure, will not be included in the output record.
6. Any output fields whose field names are not part of the input record structure are initialised to their default values.

In order to specify an output structure, both OUTPUT Fileid and OUTPUT **USING** fields must be enabled by entering "/" in their activation fields.

See the input structure USING field for description of output USING field sub-parameters and implementation of a structure on record data.

The OUTPUT USING field corresponds to the FSU parameters OUTPUT *fileid* USING *struct_name*.

`Create Output Lib`

This option field indicates that, if OUTPUT Fileid contains an as yet unallocated PDS/PDSE library DSN for library member copy, then the library should be allocated as new using the same DCB geometry and SPACE attributes as the first input library.

If this option is not selected, then the library will not be created and ZZSD353E open error message is returned.

This option field corresponds to the FSU parameter NEW.

`Append`

This option field indicates that, for non-library member copy, records are to be appended to existing data in the output file.

If this option is not selected, then the existing records will be overwritten.

This option field corresponds to the FSU parameter APPEND.

`Replace Members`

This option field indicates that, for library member copy only, existing members in the specified output library will be replaced if members of the same name are copied from the input libraries.

If this option is not selected, then existing members will not be replaced.

This option field corresponds to the FSU parameter REPLACE.

File Search/Update/Copy/Remap Output Output

Report Format

The report generated by the file Search/Update/Copy/Remap utility is a structured data file designed to be browsed (not printed) using a structure definition object (SDO) within a SELCOPY/i session.

The associated SDO is automatically generated when the utility is executed. The SDO dataset name is always the DSN of the report with a suffix of "**.SDO**".

If a report output file DSN has been specified by the user, then the data set must be an already allocated, sequential data set. If not specified, the default DSN for the report and SDO data sets is "*prefix*.FSU.Dyyyyddd.Thhmmss" and "*prefix*.FSU.Dyyyyddd.Thhmmss.SDO" respectively. The high level qualifier, *prefix*, is the value assigned to

System.UserDSNPrefix in the SELCOPY/i User INI file.

If the utility is run in the **background** as a batch job, then specification of a report output DSN is mandatory. If the job stream JCL is generated via the File Search/Update/Copy/Remap Panel, then DD statements with DISP=NEW will be generated for the specified report file DSN and SDO structure data sets, if they do not already exist.

If the utility is run in the **foreground** of a SELCOPY/i session, then the output report file is generated in storage only and is displayed and updated automatically as the utility executes. The display is refreshed every second allowing the user to view the progress and, if necessary, interrupt the execution using the Attention key.

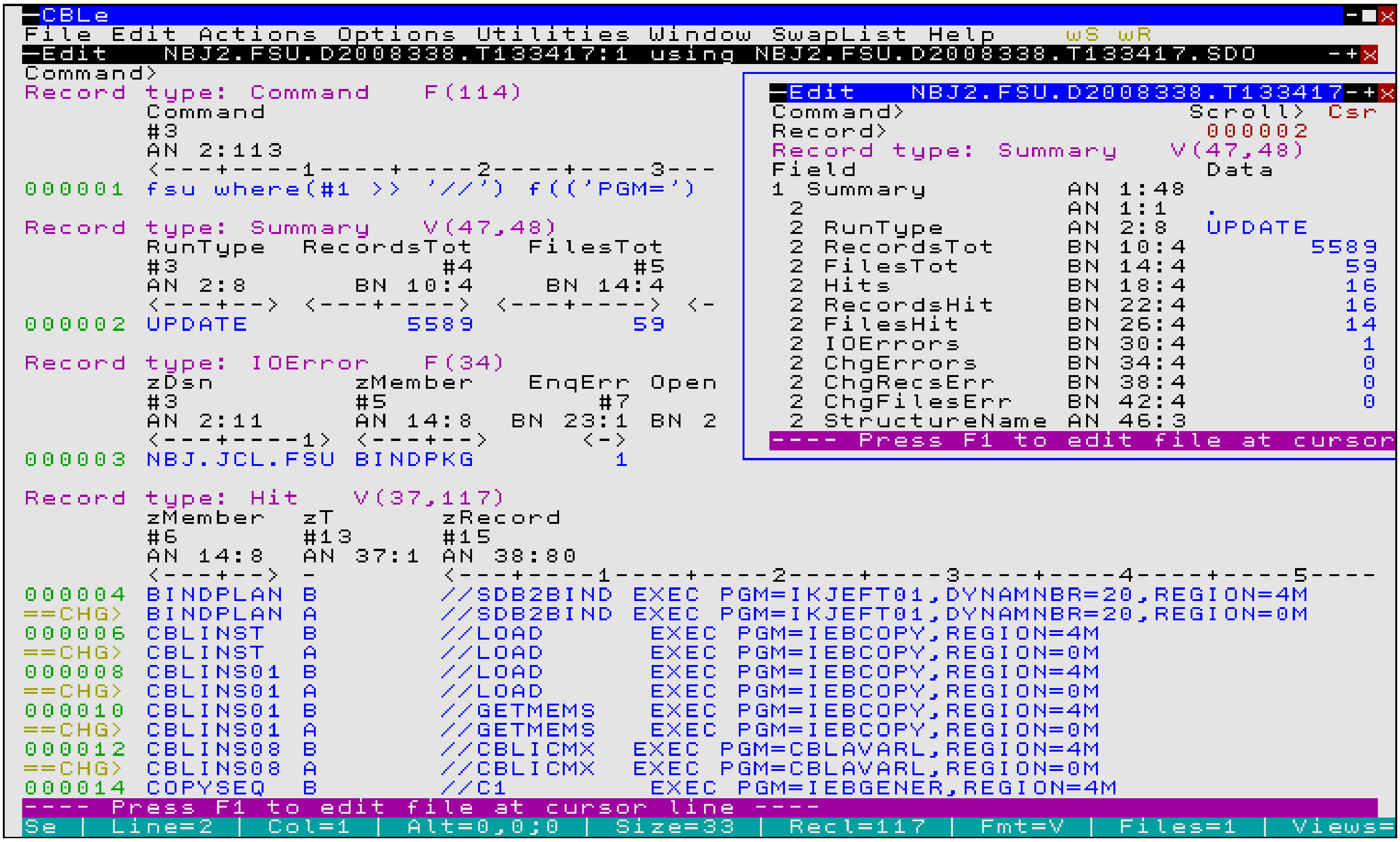
When closing an in-storage report, the user will be prompted to save the report file and accompanying SDO structure if either of the following are true:

- File update processing has occurred with the immediate UPDATE option set. Following an UPDATE, it is strongly recommended that the user save these files, so providing an audit trail and, if required, the necessary input to the File Update Undo facility which reverses changes to updated records.
- A READ or UPDATE I/O error occurs before the run has completed. This ensures that the audit trail exists even if control is not returned from the system routine.

In all other circumstances, the user will not be prompted to save a permanent copy of the in-storage report unless FSUEND is executed to close the report window. If the user chooses to save the report, the report file will be saved as a VSAM ESDS data set and the accompanying SDO as a physical sequential data set.

Following execution of the utility (in batch or in the foreground) the saved report file may subsequently be browsed from your SELCOPY/i session using any of the following methods:

- Issue the command `FSOUT` *`report_file_name`*.
- Issue the prefix command `FO` against the report DSN in a data set list or VTOC list window.
- From the File Search/Update/Copy/Remap Panel, select menu item "Reports" and press the <Enter> key or, if configured, double-click the left mouse button on the required report file.



```
-CBLe
File Edit Actions Options Utilities Window SwapList Help   wS wR
-Edit   NBJ2.FSU.D2008338.T133417:1 using NBJ2.FSU.D2008338.T133417.SDO    -+
Command>
Record type: Command   F(114)              -Edit   NBJ2.FSU.D2008338.T133417-+
       Command                             Command>                Scroll> Csr
       #3                                  Record>                  000002
       AN 2:113                            Record type: Summary   V(47,48)
       <---+----1----+----2----+----3---   Field                    Data
000001 fsu where(#1 >> '//') f(('PGM=')    1 Summary        AN 1:48
                                            2               AN 1:1  .
Record type: Summary   V(47,48)             2 RunType       AN 2:8  UPDATE
       RunType  RecordsTot   FilesTot       2 RecordsTot    BN 10:4       5589
       #3               #4         #5       2 FilesTot      BN 14:4         59
       AN 2:8      BN 10:4    BN 14:4       2 Hits          BN 18:4         16
       <---+--> <---+----> <---+----> <-    2 RecordsHit    BN 22:4         16
000002 UPDATE         5589         59       2 FilesHit      BN 26:4         14
                                            2 IOErrors      BN 30:4          1
Record type: IOError   F(34)                2 ChgErrors     BN 34:4          0
       zDsn        zMember   EnqErr Open    2 ChgRecsErr    BN 38:4          0
       #3          #5            #7         2 ChgFilesErr   BN 42:4          0
       AN 2:11     AN 14:8  BN 23:1 BN 2    2 StructureName AN 46:3
       <---+----1> <---+-->     <->        ---- Press F1 to edit file at cursor
000003 NBJ.JCL.FSU BINDPKG        1

Record type: Hit   V(37,117)
       zMember  zT      zRecord
       #6       #13     #15
       AN 14:8  AN 37:1 AN 38:80
       <---+--> -       <---+----1----+----2----+----3----+----4----+----5----
000004 BINDPLAN B       //SDB2BIND EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=4M
==CHG> BINDPLAN A       //SDB2BIND EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=0M
000006 CBLINST  B       //LOAD      EXEC PGM=IEBCOPY,REGION=4M
==CHG> CBLINST  A       //LOAD      EXEC PGM=IEBCOPY,REGION=0M
000008 CBLINS01 B       //LOAD      EXEC PGM=IEBCOPY,REGION=4M
==CHG> CBLINS01 A       //LOAD      EXEC PGM=IEBCOPY,REGION=0M
000010 CBLINS01 B       //GETMEMS   EXEC PGM=IEBCOPY,REGION=4M
==CHG> CBLINS01 A       //GETMEMS   EXEC PGM=IEBCOPY,REGION=0M
000012 CBLINS08 B       //CBLICMX  EXEC PGM=CBLAVARL,REGION=4M
==CHG> CBLINS08 A       //CBLICMX  EXEC PGM=CBLAVARL,REGION=0M
000014 COPYSEQ  B       //C1        EXEC PGM=IEBGENER,REGION=4M
---- Press F1 to edit file at cursor line ----
Se | Line=2 | Col=1 | Alt=0,0;0 | Size=33 | Recl=117 | Fmt=V | Files=1 | Views=
```

Figure 31. File Search/Update/Copy/Remap Output Output.

In the FSU output displayed in Figure 31., the user has hit the PF2 key on the Summary record, to display the record's fields in single view, and also executed the following SDE SELECT command to restrict the fields displayed in records of record type Hit:

```
SELECT   zMember, zT, zRecord   FROM Hit
```

The FSU report output consists of 5 record types: 1 Command record, 1 Summary record, 0 or more Hit records, 0 or more IOError records and 0 or more Alias records.

Record Type: Command

Contains information of the FSU command stream use to execute the File Search/Update/Copy/Remap utility.

`Timestamp`
The date and local time at which the FSU command was executed to generate this report.

`Command`
A character field containing the FSU command executed (directly or via the File Search/Update/Copy/Remap Panel). The Command record is located at the top of the output report.

Record Type: Summary

Contains statistical fields providing totals for the File Search/Update/Copy/Remap Output execution as follow:

`RunType`
Describes the type of execution of the File Search/Update/Copy/Remap Output utility and so governs the format of the Hit records.

Operation	RunType
Unformatted/Formatted Search	FIND
Unformatted/Formatted Update	UPDATE or NOUPDATE
Unformatted/Formatted Copy or Remap	COPY

For Update operations, RunType is "NOUPDATE" if update has been suppressed by the FSU command NOUPDATE (default) parameter or "Immediate UPDATE" has been deactivated in the utility panel.

`RecordsTot`
The total number of input records successfully read from all selected files.

`FilesTot`
The total number of files that match the supplied fileid mask(s).

`Hits`
For RunType "FIND", this is the total number of occurrences of the FIND search string(s) found within the input records that satisfy the logical combination of FIND conditions. e.g. `FIND ( ('A') AND ('B') )` will increment the Hit total for every occurrence of 'A' and 'B' in records that contain both 'A' and 'B'.

For RunType "NOUPDATE" and "UPDATE", this is the total number of occurrences of the CHANGE search string(s) found within the selected input records (i.e. input records that satisfy the VIEW, WHERE and/or FIND criteria.)

`RecordsHit`
For RunType "FIND", this is the total number of input records that satisfy the supplied WHERE clause and/or FIND search string criteria.

For RunType "NOUPDATE" and "UPDATE", this is the total number of selected input records that satisfy the CHANGE search string criteria.

RecordsHit corresponds with the number of input records that are of the record type "Hit Records" and so are displayed in the report output.

`FilesHit`
The total number of files that contains at least one record that includes a hit.

`RemapErrors`
For Formatted Remap operations only, the total number of files for which an error has occurred on attempting to remap source fields of one data type to target fields of a different data type.

`IOErrors`
The total number of files for which I/O errors that have occurred during execution. An IOError record is displayed for each

`ChgErrors`

The total number of CHANGE errors. i.e. the total number of occurrences of a CHANGE search string, within all selected records, that cannot be updated with the CHANGE replace string.
For RunType "FIND", this value is aways 0 (zero).

`ChgRecsErr`

The total number of selected input records for which a CHANGE error has occurred.
For RunType "FIND", this value is aways 0 (zero).

`ChgFilesErr`

The total number of files that contain at least one record for which a CHANGE error has occurred.
For RunType "FIND", this value is aways 0 (zero).

`StructureName`

The DSN and member name of the structure (SDO) specified on the USING field/parameter for Formatted File Search/Update/Copy/Remap Output.
For Unformatted File Search/Update/Copy/Remap Output, this field value is always blank.

Record Type: Hit

The format of the Hit records depend on the Summary record "RunType" field, as follows:

1. For RunType "FIND" or "COPY" where search criteria have been specified, this field displays every record that satisfies all the specified search criteria. i.e. the VIEW record type (for Formatted File Search/Update/Copy/Remap Output), the WHERE clause and the FIND search string(s). Note that no CHANGE operation has been specified.

2. For RunType "NOUPDATE", "UPDATE" or "COPY" where a CHANGE operation has been specified, this field displays a pair of records for every input record that satisfies the CHANGE arguments. The first record of the pair displays the original, unaltered record data, the second displays the record data after the CHANGE operation(s) have been executed.

   Depending on whether the CHANGE operation(s) are successful, the prefix area of the line displaying the updated record will contain the line flag "==CHG>" or, if an error has occurred, "==ERR>".

   For update operations with NOUPDATE in effect, this allows the user the opportunity to check the changed data before re-running the utility with UPDATE to action the changes.

The Hit record type contains 2 group fields, "z" and "zRecord", where "z" includes information fields relating to the record, and "zRecord" includes all the record data field(s). The Hit record type has been designed this way so that the user can suppress or include all fields within either field group by specifying the group field name as the argument of a SELECT command. e.g. `SELECT zRecord` will display only the field data and suppress the information fields.

The Hit record type field structures are as follow:

`z`

The structure including all the information fields.

`zDsn`

The DSN (or HFS path name) containing the reported record.

`zMember`

The PDS/PDSE member containing the reported record.
This field is only present if at least one PDS/PDSE data set is included by the input fileid mask(s). For non-PDS/PDSE data sets, this field contains blanks.

`zRecNo`

The record number of the record within the data set.

`zHitNo`

The hit count number of the record within the data set. The "zHitNo" field is incremented by one for each new record within the data set that satisfies the search criteria for the particular RunType. The "zHitNo" count is reset to zero for each new input data set. Use `WHERE zHitNo=1` to display a list of all data sets (and library members) containing at least one hit.

`zLrecl`

The record length of the record within the data set.

`zHits`

The total number of occurrences of the search string(s) within the record.

For all RunType "FIND"/"COPY" Hit records or RunType "NOUPDATE"/"UPDATE"/"COPY" Hit records with "zT" field flag set to "B", this is the number of FIND search string occurrences.

For all RunType "NOUPDATE"/"UPDATE"/"COPY" Hit records with "zT" field flag set to "A", this is the number of CHANGE search string occurrences.

`zErrs`

For RunType "NOUPDATE", "UPDATE" and "COPY" involving a CHANGE operation, the total number of occurrences of a CHANGE search string within the record that cannot be updated with the CHANGE replace string.

For RunType "FIND", this field is omitted.

`ZT`

Included only for RunType "NOUPDATE", "UPDATE" and "COPY" involving a CHANGE operation, this field displays the record image flag which may be one of the following:

B Indicates that the record data that follows represents the record data **Before** the CHANGE operation(s) are applied.

A Indicates that the record data that follows represents the record data **After** the CHANGE operation(s) are applied.
Note that the record data will be unchanged if the values in the fields "zHits" and "zErrs" are equal.

For RunType "FIND", this field is omitted.

`zRecord`

The structure including all the record data fields.

For Unformatted File Search/Update/Copy/Remap Output, the "zRecord" field contains the unexpanded record data as a single character field of length equal to the record length.

For Formatted File Search/Update/Copy/Remap Output, the "zRecord" field contains the expanded record data mapped with the field names defined by the record type (RTO).

Record Type: IOError

Contains information relating to an I/O error that has occurred when opening, reading or updating the file. IOError records are located amongst the Hit records, as I/O errors are encountered.

`zDsn`

The DSN (or HFS path name) for which the I/O error occurred.

`zMember`

The PDS/PDSE member for which the I/O error occurred.
This field is only present if the at least one PDS/PDSE data set is included by the input fileid mask(s). For non-PDS/PDSE data sets, this field contains blanks.

`EnqErr`

1 if the error occurred when attempting to obtain an exclusive SPFEDIT ENQ for UPDATE on the file, otherwise 0.

`OpenErr`

1 if the error occurred when attempting to open the file, otherwise 0.

`ReadErr`

1 if the I/O error occurred when attempting to read a block of data from the file, otherwise 0.

`UpdErr`

1 if the I/O error occurred when attempting to re-write (update-in-place) a record to the file, otherwise 0.

`OutErr`

1 if the I/O error occurred when attempting to write a block of data to the OUTPUT file, otherwise 0.

`RecordsRead`

The number of records successfully read before the I/O error occurred.

`RecordsUpd`

The number of records successfully updated before the I/O error occurred.

Record Type: Alias

Contains information relating to aliases of library members. An Alias report record is generated for every member and member alias that contains a hit for WHERE/FIND search criteria or a CHANGE operation.

When processing PDS/PDSE library members, member records may be searched or changed having been accessed via the original member name or a member alias name. Once the member records have been processed, they will not be processed again via another alias name or their member name.

`zDsn`
The DSN of the PDS/PDSE library.

`zMember`
The PDS/PDSE member or member alias name.

`zAliasOf`
The PDS/PDSE member name for which this library entry is an alias.

`zHitRef`
The member or alias name for which associated "Hit" report records contain the search or change results applicable to the alias or member name identified by this "Alias" report record.

Record Type: Record

Contains a single, variable length character field "Record" displaying any error messages that have been generated by the File Search/Update/Copy/Remap utility.

Function Keys

<PF1>	Display context sensitive help.
<PF2>	Display the report record in a new window in single format (vertical) view. In single format view, use **<PF10>**/**<PF11>** to display the previous/next report record respectively.
<PF4>	Display the SDE Edit/Browse utility menu. This includes show and hide of report records based on their type, and alter the display of report record fields.
<PF6>	Applicable to report records of record type Hit or IOError only, <PF1> edits the file(s), referenced by "zDSN" and "zMember" fields in the focus report record, and scrolls directly to the reported record. If the input records were formatted using an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to map the record data fields, then SDE EDIT is performed using the existing or generated SDO structure. For unformatted records, the SELCOPY/i text editor is used to edit the records.

File Update Undo

Overview

The File Update Undo utility (FSUUNDO) allows the user to restore updated records from any accidental or erroneous execution of the File Search/Update/Copy/Remap utility (FSU) where update of record data has occured.

When FSU is executed to change and immediately UPDATE data set records, the original record data, before execution of the change operation(s), is recorded in the FSU output report data set. FSUUNDO uses these report records as part of its processing and so will only operate successfully if the FSU report data set exists.

Therefore, it is strongly recommended that, when prompted on exit from the report data set, users elect to save the report and its accompanying SDE structure (SDO), for audit purposes and also for subsequent execution of FSUUNDO if required.

FSU report output reflecting FIND or NOUPDATE run types need not be saved. If used as input to FSUUNDO, FSU report output of run type FIND will return an error.

FSUUNDO generates a SELCOPY control statements that performs the following:

1. Input the FSU report records.
2. For each "Hit" record "Before" and "After" pair reported in the FSU output, identify the DSN, PDSE(E) member name (if applicable) and record number at which the updated record may be found.
3. Sequentially read records from the data set or PDS(E) member until the required record number is found.
4. Verify that the input record matches the record "After" data reported in the FSU output.
5. Optionally UPDATE the data set record with the record "Before" data reported in the FSU output, thus restoring the record to its original status.
6. Generate an entry in the File Update Undo report.
7. Repeat all steps until all "Hit" record pairs in the FSU output report have been processed.

Error checking is also performed for conditions which include record not found or containing unexpected data. These conditions are reported in the FSUUNDO output report.

This SELCOPY job may be executed in the Foreground (TSO) or displayed as a JCL job, suitable for submission to batch. Using either method, options exist to generate an Expanded or Terse report, an optional Diagnostic SELCOPY execution report and, most importantly, options to Verify or Update changed records.

It is strongly recommended that users execute FSUUNDO with options VERIFY (the default) and EXTENDED, and then review the FSUUNDO output report before executing FSUUNDO again with option UPDATE.

Beware that record data that has been changed between the time of execution of the original File Search/Update/Copy/Remap job and this execution of FSUUNDO, will not be updated but will be reported as an error. Processing will continue with input of the next "Hit" record pair. Records that have already been restored as a result of a previous execution of FSUUNDO UPDATE, will be reported as a match and no error returned.

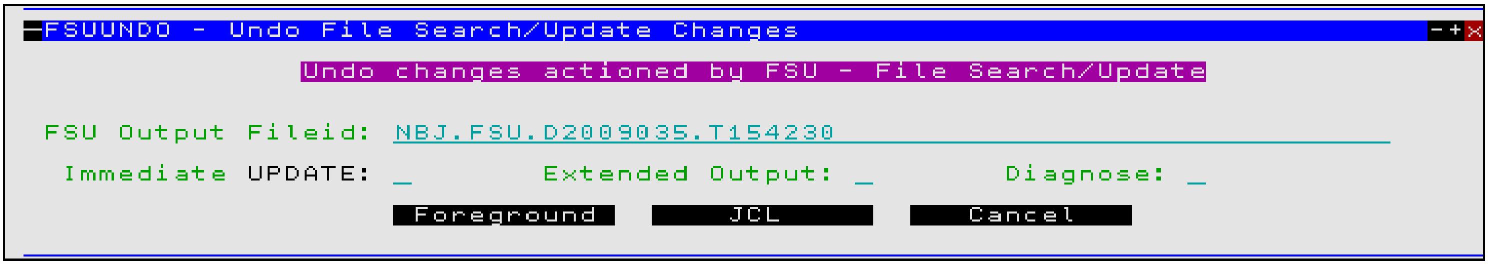
If SELCOPY ends with a return code other than 0 (zero - successful execution, no error conditions) or 112 (errors condition(s) detected), then re-run with option DIAGNOSE to establish the cause of the SELCOPY error.

The most likely cause of an unxpected return code will be if a selection (run) time error (RC=44) has occured. Usually caused by an OPEN error for an input data set (e.g. if an exclusive ENQ already exists for the data set.) In this event, SELCOPY processing ends immediately and all data sets opened by SELCOPY are automatically closed.

File Update Undo Panel

The File Update Utility Undo panel view may be started by executing command FSUUNDO from the command line of any window.

Field options may be selected or de-selected by entering a non-blank or blank character respectively in the option field.



FSUUNDO - Undo File Search/Update Changes

Undo changes actioned by FSU - File Search/Update

FSU Output Fileid: NBJ.FSU.D2009035.T154230

Immediate UPDATE: _ Extended Output: _ Diagnose: _

Figure 32. FSUUNDO - File Update Utility Undo Panel.

File Update Undo Output

Report Format

Report output is generated on every execution of the File Update Undo utility (FSUUNDO).

If FSUUNDO is executed with parameter BACKGROUND (JCL) to generate JCL output, then the output report is written to SYSPRINT when the job is submitted. By default, SYSPRINT is allocated to SYSOUT=*. Furthermore, if DIAGNOSE parameter was specified, then the SYSPRINT output will also contain diagnostic information for the SELCOPY run before and after the printed report output.

If FSUUNDO is executed with parameter FOREGROUND to execute in TSO, then an output report data set is opened in a CBLe edit view and report records are inserted. The DSN of this data set is equal to the FSU report DSN with the additional low level qualifier "UNDOV" for VERIFY reports, or "UNDO" for UPDATE reports. e.g.

```
NBJ2.DEV.FSU.D2008346.T162607.UNDO
```

If this data set already exists, then the report records will be appended to the existing report data and the edit display positioned at the start of the latest report output. On exit of this data set the user will be prompted to save and, if necessary allocate, the data set with suitable space attributes.

Report Fields

Dataset
The up to 44 character DSN of the data set or PDS/PDSE processed.

Member
For PDS/PDSE libraries only, the name of the member being processed.

RecordNumber
The record number at which an FSUUNDO error has occurred.
For EXTENDED output, this field also contains the record numbers of records that have been successfully updated.

Message Text
Message indicating success or failure to locate and update records referenced by the FSU report. All possible messages are as follow:

```
= = File Updated = =
= = Member Updated = =
```

One or more records within the reported data set or PDS/PDSE member were successfully updated.

For EXTENDED output only, this message is repeated for each record that has been successfully updated. Also, up to 100 bytes of the record data before and after the update is printed on the report lines that follow.

```
## Member not found ##
```

The member name within the PDS/PDSE library referenced by the FSU report line, no longer exists. The member has been deleted or renamed.
Return Code 112 is set and the error count incremented by one for each missing member.

```
## Record not found ##
```

The record number of the data set or member record referenced by the FSU report line, no longer exists. This message is repeated for each missing record within a data set or member.

For EXTENDED output only, up to 100 bytes of the expected record data is also printed on the report lines that follow.
Return Code 112 is set and the error count is incremented by one for each missing record.

```
### Data Mismatch ###  --WARNING--
```

The record data at the required record number within the data set or member, does not match the record data as it was following the FSU update. (i.e. the data has been altered since the FSU update was executed.) This message is repeated for each mis-matching record within a data set or member.

For both EXTENDED and TERSE output, up to 100 bytes of the found record data, followed by up to 100 bytes of the expected record data, is also printed on the report lines that follow.
Return Code 112 is set and the error count is incremented by one for each missing record.

```
<<< Already Undone >>>
```

This message is generated for EXTENDED report output only. TERSE report output does not report instances where no record update is required.

The record data at the required record number within the data set or member, matches the record data as it was prior to the FSU update. (i.e. the record has already been restored.)

This message is repeated for each already restored record within a data set or member and the records already undone count is incremented by one. Up to 100 bytes of the record data is also printed on the following report lines.

Report Data

For EXPANDED output or where a data mismatch error has occurred, record data is reported in the FSUUNDO output.

The data lines are preceded by a brief description of the record. Where two contrasting records are to be displayed, the description line also reports the position within the record data of the first difference found. Furthermore, if this difference occurs beyond the record description and within the first 100 bytes, a marker ">*<" is positioned above the mis-matching character.

The record data follows the description line for a length equal to the lesser of the record length value and 100. The data is displayed in both character and vertical hexadecimal notation. (Equivalent to SELCOPY's TYPE=B print output.)

A scale line is written following the record data.

```
Record Found:  (1st Diff at Pos    049)        >*<
EDTFCLR0 0720 MEMFLSTE >>>>>>>>>>AZZElement     Test                       3 117ADA10 117BFAC8 00000..
CCECCDDF4FFFF4DCDCDEEC46666666666CEEC98989A44444E8AA44444444444444444444444F4FFFCCCFF4FFFCCCCF4FFFFF
54363390007200454632350EEEEEEEEEE1995354553000003523000000000000000000000003011714110011726138000000

Record Expected:
EDTFCLR0 0720 MEMFLSTE >>>>>>>>>>AZZElement                                3 117ADA10 117BFAC8 00000..
CCECCDDF4FFFF4DCDCDEEC46666666666CEEC98989A44444444444444444444444444444444F4FFFCCCFF4FFFCCCCF4FFFFF
54363390007200454632350EEEEEEEEEE19953545530000000000000000000000000000000003011714110011726138000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8....,....9....,...10
```

Figure 32. Record Data Output.

Summary Block

The summary block reports the totals of:

1. Records undone
2. Records already undone
3. Files updated
4. Errors

Where FSUUNDO errors have occured, the Errors total has the additional eye-catcher "###" following.

Sample Terse Report Output

```
------------------------------------- Start of FSUUNDO Report ----------------------------------------


  Output From:  FSUUNDO  NBJ2.DEV.FSU.D2008357.T132545  FOREGROUND  VERIFY  TERSE


  ** Verify Only - No data sets will be updated. **



-------------------------------------------- -------- ------------         ----------------   --------
Dataset                                      Member   RecordNumber         2008/12/22 13:27   PAGE   1
-------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                  CBLINST              = = Member Updated = =
NBJ.JCL.FSU                                  CBLINS01             = = Member Updated = =
NBJ.JCL.FSU                                  CBLINS01         188  ## Record not found ##
NBJ.JCL.FSU                                  EQAWIVP1             = = Member Updated = =
NBJ.JCL.FSU                                  IEBCOPY1          51   ### Data Mismatch ###  --WARNING--

Record Found:  (1st Diff at Pos    046)     >*<
//GETMEMS   EXEC PGM=IEBCOPY,REGION=0M       ## Get ###
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD4444444774C8A4777444444444444444444444444
11753454200057530774E9523678B957965E040000000BB07530BBB0000000000000000000000000

Record Expected:
//GETMEMS   EXEC PGM=IEBCOPY,REGION=0M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11753454200057530774E9523678B957965E04000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8

NBJ.JCL.FSU                                  IEBCOP02             = = Member Updated = =

     --------------------------------------
     Records undone:                      4
     Records already undone:              0
     Files   updated:                     5
     Errors:                              2###
     --------------------------------------

  ** Verify Only - No data sets have been updated. **


  *** Warning:  SELCOPY execution ended with RC=112 ***

  One or more data sets have been altered between the time of execution of
  the original File Search/Update job and this execution of FSUUNDO.
```

Figure 32. FSUNDO TERSE Output with VERIFY.

Sample EXTENDED Output

```
------------------------------------- Start of FSUUNDO Report ----------------------------------------


  Output From:  FSUUNDO  NBJ2.DEV.FSU.D2008357.T132545  FOREGROUND  UPDATE  EXTENDED




-------------------------------------------- -------- ------------         ----------------   --------
Dataset                                      Member   RecordNumber         2008/12/22 13:35   PAGE   1
-------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                  CBLINST           42  = = Member Updated = =

Record Old:  (Changed data starts at Pos    037)
//LOAD      EXEC PGM=IEBCOPY,REGION=4M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11361400000057530774E9523678B957965E44000000000000000000000000000000000000000000

Record New:
//LOAD      EXEC PGM=IEBCOPY,REGION=0M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11361400000057530774E9523678B957965E04000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8




-------------------------------------------- -------- ------------         ----------------   --------
Dataset                                      Member   RecordNumber         2008/12/22 13:35   PAGE   1
-------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                  CBLINS01         119  = = Member Updated = =

Record Old:  (Changed data starts at Pos    037)
//LOAD      EXEC PGM=IEBCOPY,REGION=4M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11361400000057530774E9523678B957965E44000000000000000000000000000000000000000000

Record New:
//LOAD      EXEC PGM=IEBCOPY,REGION=0M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11361400000057530774E9523678B957965E04000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8

NBJ.JCL.FSU                                  CBLINS01         188  ## Record not found ##

Record Expected:
//GETMEMS   EXEC PGM=IEBCOPY,REGION=4M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11753454200057530774E9523678B957965E44000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8




-------------------------------------------- -------- ------------         ----------------   --------
Dataset                                      Member   RecordNumber         2008/12/22 13:35   PAGE   1
-------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                  EQAWIVP1         124  = = Member Updated = =

Record Old:  (Changed data starts at Pos    046)
//COMPILE EXEC PGM=IGYCRCTL,PARM=(''),REGION=4M
66CDDDCDC4CECC4DCD7CCECDCED6DCDD747756DCCCDD7FD444444444444444444444444444444444
1136479350575307 74E97839333B7194EDDDDB957965E44000000000000000000000000000000000

Record New:
//COMPILE EXEC PGM=IGYCRCTL,PARM=(''),REGION=0M
66CDDDCDC4CECC4DCD7CCECDCED6DCDD747756DCCCDD7FD444444444444444444444444444444444
11364793505753077 4E97839333B7194EDDDDB957965E04000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8

NBJ.JCL.FSU                                  EQAWIVP1         203  = = Member Updated = =

Record Old:  (Changed data starts at Pos    043)
//LKED    EXEC PGM=HEWL,PARM='MAP',REGION=4M
66DDCC4444CECC4DCD7CCED6DCDD77DCD76DCCCDD7FD444444444444444444444444444444444444
11325400005753077 4E8563B7194ED417DB957965E44000000000000000000000000000000000000

Record New:
//LKED    EXEC PGM=HEWL,PARM='MAP',REGION=0M
66DDCC4444CECC4DCD7CCED6DCDD77DCD76DCCCDD7FD444444444444444444444444444444444444
11325400005753077 4E8563B7194ED417DB957965E04000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8




-------------------------------------------- -------- ------------         ----------------   --------
Dataset                                      Member   RecordNumber         2008/12/22 13:35   PAGE   1
-------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                  IEBCOPY1          51   ### Data Mismatch ###  --WARNING--

Record Found:  (1st Diff at Pos    046)     >*<
```

```
//GETMEMS   EXEC PGM=IEBCOPY,REGION=4M       ## Get ###
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD4444444774C8A47774444444444444444444444444
11753454200057530774E9523678B957965E440000000BB07530BBB0000000000000000000000000

Record Expected:
//GETMEMS   EXEC PGM=IEBCOPY,REGION=4M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11753454200057530774E9523678B957965E44000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8




-------------------------------------------- -------- ------------         ----------------   --------
Dataset                                      Member   RecordNumber         2008/12/22 13:35   PAGE   1
-------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                  IEBCOP02           7  = = Member Updated = =

Record Old:  (Changed data starts at Pos    037)
//UNLOAD    EXEC PGM=IEBCOPY,REGION=4M
66EDDDCC4444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11453614000057530774E9523678B957965E44000000000000000000000000000000000000000000

Record New:
//UNLOAD    EXEC PGM=IEBCOPY,REGION=0M
66EDDDCC4444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11453614000057530774E9523678B957965E04000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8


     --------------------------------------
     Records undone:                      5
     Records already undone:              0
     Files   updated:                     4
     Errors:                              2###
     --------------------------------------


  *** Warning:  SELCOPY execution ended with RC=112 ***

  One or more data sets have been altered between the time of execution of
  the original File Search/Update job and this execution of FSUUNDO.
```

Figure 32. FSUNDO EXTENDED Output with UPDATE.

Compare Files/Libraries Menu (=7)

The Compare Files/Libraries Menu panel (ZZSGCOMP) is an interactive panel window opened on selection of option 7. in the SELCOPY/i Primary option menu.

SELCOPY/i data object compare features can be accessed via this panel.

Options

1 Compare Files	COMPF - Compare Files
2 Compare Libraries	COMPL - Compare Libraries

Compare Files (=7.1)

Overview

The Compare Files utility (COMPFILE) provides a set of both basic and extended features that allow the user to compare records in **NEW** and **OLD** versions of a file.

Basic features include:

- Specify the start record.
- Restrict the number of records compared.
- Restrict the number of differences to be reported.
- Restrict the comparison to a specific area of the file records.
- Strip trailing characters prior to record compare.

Extended features include:

- All basic feature options but with separate specifications for the NEW and OLD files where sensible.
- Apply a **structure** (copybook) overlay to map records, and optionally restrict the comparison to specified record-types and/or named fields. This is known as a **formatted compare**.
- Control how re-synchronisation of record pairs should occur following detection of an inserted or deleted record.
- For formatted or unformatted compare, specify **key** segments (at the record-type level) that allow the utility to identify **synchronised pairs** of records.
- Formatted compare supports application of different structures to the NEW and OLD files, with comparison restricted to only those fields that exist in both structures. This allows comparison of NEW and OLD file records where corresponding fields are at different locations within the records and maybe of different data-type or length.

Following Compare Files execution, report output is generated in a structured format suitable for presentation to the user in an SDE window view. To generate this report output and in order to perform advanced record selection and field compare, COMPFILE utilises functions and features provided by the structured data environment (SDE). Therefore, the COMPFILE utility is only available to users who have a licensed version of SELCOPY installed and operational on their system.

Source File Types

COMPFILE can process records from any combination of the following file types:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.
- (DB2 Tables planned but not yet supported).

Output Report

The report generated by the compare files utility is a **structured data file**. This is designed to be browsed (not printed) from within a SELCOPY/i session using a structure definition file (SDO) which is also generated automatically during execution of the compare.

Following execution of the compare utility, records are flagged as being **matched** or as having been **inserted**, **deleted** or **changed**.

Matched
Records that exist in both the NEW and OLD files forming a synchronised record pair for which the compared data is unchanged (matches).

Inserted
Records that previously did not exist in the old file and so have been inserted into the NEW file.

Deleted
Records that no longer exist in the NEW file and so have been deleted from the OLD file.

Changed
Records that exist in both files forming a synchronised record pair in which the compared data has been changed (i.e does not match).

Determination of synchronised record pairs is achieved by the compare file utility using record synchronisation techniques.

In order to improve readability, the report of consecutive records flagged as having been deleted are grouped together, and similarly for records flagged as having been inserted.

See Compare Files Output for a detailed description of the generated output report.

Unformatted Compare

Unformatted compare is the most commonly used format for **text** files containing unformatted records.

By definition, unformatted compare operates on records without application of a structure (SDO) or COBOL/PL1 copybook to format record data. i.e. each record is treated as a single character string.

Basic Unformatted Compare
Basic unformatted compare specifically relates to unformatted compare where selected options apply to **both** files involved in the compare operation. These options are:

◊ The compare data start position within the record.
◊ The compare data length.
◊ The trailing character to be stripped before comparing the data.
◊ The first record to be compared. (Nominated by record number, key or RBA.)
◊ The number of records to be compared.

Furthermore, record synchronisation technique employed is restricted to 1-TO-1 or read-ahead with a read-ahead limit of 100 records and read-ahead matching record count of 1.

The Compare File utility panel view and input field options relating to basic unformatted compare are described in *"Basic Unformatted Compare Panel"*.

Extended Unformatted Compare
Extended unformatted compare allows specification of the same options as basic unformatted compare but with potentially different values for each of the two files in the compare operation. In addition to this, extended unformatted compare allows specification of the following:

◊ Record synchronisation techniques Sorted Keyed and Unsorted Keyed which involves specification of key segments.
◊ For read-ahead record synchronisation, non-default values for limit and matching record count. Also the option to allow synchronisation on blank records.
◊ The option to perform case-insensitive compare.
◊ Report output options to exclude display of changed, inserted and/or deleted records. Also allows specification of a non-default report file DSN.
◊ Output file DSNs into which to copy records flagged as being matched, changed, inserted and/or deleted. A separate data set name may be specified for NEW and OLD file records that are attributed these flags.

The Compare File utility panel views and input field options relating to extended unformatted compare are described in *"Extended Unformatted Compare Panel"*.

Formatted Compare

More advanced than unformatted compare, formatted compare is invoked where an SDE structure (SDO), COBOL or PL1 copybook overlay is specified to map record data fields for use in the compare files operation.

Records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records, and fields to be compared.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO) defined in the specified or generated SDO and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

Formatted compare may be selected via the Compare File utility panel by first selecting **Extended options** from the Compare Files Basic Options view. Compare File utility panel views and input field options relating to formatted compare are described in *"Formatted Compare Panel"*.

Hierarchical Compare

Hierarchical compare is not selected explicitly but is implied when both of the following conditions are true:

1. Formatted compare is used incorporating records assigned to different record types in the SDO.
2. KEY synchronisation is performed with key segments specified as formatted record field names or field reference numbers.

The compare files command, COMPFILE, generated by the dialog panel or entered manually by the user, specifies synchronisation key fields for one or more record types in the specified SDO. The order in which these record types occur in the COMPFILE command also define the levels of record type hierarchy. i.e. The record type synchronisation key definition occurring first identifies the level-1 (highest level) record type, the second definition identifies the level-2 (level-1 child) record type, etc.

Record types with no synchronisation key are the lowest level in the record type hierarchy, i.e. rated lower than any record type that has been defined with a synchronisation key.

Hierarchical compare is sensitive to the level of record type assigned to a record. All records that immediately follow the current record which are assigned record types lower in the record type hierarchy than that of the current record, are treated as being descendants of the current record. These records are grouped with the current record so that record synchronisation does not exceed the bounds of the current hierarchical record group.

This type of compare ensures that only record pairs that belong to the same hierarchical parent record pair can be synchronised.

For details on the synchronisation criteria and the synchronisation process, see *"Hierarchical Key Synchronisation"*.

Record Synchronisation

The process of comparing records requires that a pair of records, one from NEW file and one from the OLD file, are first synchronised before the data compare can take place.

Terminology

Before learning the different methods of synchronisation, a user should become familiar with the following terms:

Record Pair
Refers to one record from the NEW file and one from the OLD file.

Synchronised Record Pair
A record pair for which the records satisfy synchronisation criteria and so are eligible to be compared.

See the different record synchronisation techniques which identify synchronisation criteria.

Current Record Pair
Identifies the record pair for which synchronisation will occur.

The current record pair is usually the next record pair read following processing of a synchronised record pair. Unless End-of-Data is encountered, re-synchronisation processing will invariably change one of the records selected for the current record pair.

Compare Data
The compare data is identified as belonging to those areas of the record for which the compare operation will be actioned. Record data that falls outside these areas is not compared.

Unformatted compare data areas are specified via the Compare File panel "`Compare Position/Length:`" fields or the COMPFILE command STARTCOL/COMPARELEN parameters (or their derivatives).
Formatted and Hierarchical compare data areas are specified via the Compare File sub-panel *"Select Field Names to Compare"* or the COMPFILE command SELECT *field_col* parameter.

By default, compare data is all data in the record.

Keyed Record
Applicable only to file compare with KEY synchronisation, a keyed record is a record for which one or more compare files synchronisation key field segments have been defined.

For unformatted or formatted compare where key segments are specified using fixed field positions and lengths, this applies to all records in the NEW and OLD files.

For formatted compare where key segments are specified using formatted record field names, this applies only to records in the NEW and OLD files that are assigned record types for which a key has been defined.

Unkeyed Record

Applicable to compare of formatted or unformatted files, unkeyed records are records not defined a compare files synchronisation key field.

For hierarchical compare, an unkeyed record is a record assigned a record type for which no synchronisation key has been defined. Unkeyed records are of record types which are lowest level in the record type hierarchy.

Overview

Following input of a record pair, the compare files utility must use the rules defined by the specified synchronisation technique to determine whether the current record pair is a synchronised record pair.

If the current record pair is not a synchronised record pair, then record synchronisation is performed for one or both of the records in the current record pair. With the exception of 1-TO-1 synchronisation, this involves reading additional records from the NEW and/or OLD files in order to synchronise with a record in the current record pair.

If a synchronised record pair is established, then the compare of record data is performed so that the records in the record pair are flagged as being either **matched** or **changed**. Following the compare, the next record pair is read sequentially from the NEW and OLD files and the synchronisation process begins again.

Records in the NEW file that are been skipped as a result of record synchronisation are flagged as having been **inserted**. Similarly, skipped records in the OLD file are flagged as having been **deleted**.

The possible record synchronisation techniques are described below.

1-TO-1 Synchronisation

All current record pairs are synchronised record pairs so no attempt will ever be made to synchronise records. Mismatching record pairs are flagged as a record change and only additional records in the NEW or OLD files will be flagged as being inserted or deleted respectively.

This also applies to formatted compare so even records of different record types are considered to be a synchronised record pair.

This technique corresponds to the COMPFILE command parameter `SYNC 1TO1`.

Read-Ahead Synchronisation

Read-Ahead synchronisation is suitable where the NEW and OLD files are predominantly comprised of equal records, although some may have been updated, inserted and/or deleted.

Read-ahead synchronisation criteria requires that the compare data within the records that constitute the current record pair must match.
Note that, for formatted compare, read-ahead synchronisation will consider only record pairs of the same record type as being potential synchronised record pairs.

If the current record pair is not a synchronised record pair, then records are read sequentially from one or both files in order to identify a record that constitutes a synchronised record pair with one of the records in the current record pair.

Read-ahead synchronisation for unformatted or formatted compare proceeds as follows:

1. From the OLD file, a specified number of records are read as defined by the read-ahead limit (default 100) until one is found that matches the NEW file record in the current record pair. If found, this record becomes the OLD file record in the current record pair.

   The read-ahead match value (default 1) identifies the total number of consecutive matching record pairs that must exist, starting at the current record pair, before the current record pair can be considered a potential synchronised record pair.

   By default, matching pairs of blank records are **not** included in the read-ahead match count of consecutive matching record pairs. This avoids erroneously synchronising on blank records within inserted or deleted blocks of records. If blank records are to be included in the read-ahead synchronisation, then this may be set in the compare files panels or by specifying parameter SYNCONBLANK on the COMPFILE line command.

   If the read-ahead number of consecutive matching pairs is not satisfied, then the read-ahead synchronisation process continues for the records in the OLD file until End-of-Data or the read-ahead limit is reached.
2. Employing the same conditions, the read-ahead process is repeated for records in the NEW file in order to identify a potential synchronisation record pair with the OLD file record in the original current record pair.
3. If only one possible synchronised record pair is identified, then this becomes the new synchronised record pair.

If read-ahead in the OLD and NEW files identify different potential synchronised record pairs, then synchronisation will occur at the record pair with the fewer number of records between it and the current record pair. If this number of intervening records is equal for both record pairs, then synchronisation occurs for the synchronised record pair found in the read-ahead of the NEW file.

4. If no potential synchronised record pairs are identified, then the current record pair constitutes a record mismatch corresponding to a NEW file record insert and an OLD file record delete. The compare operation continues at the next record pair read sequentially from both files.

This synchronisation technique corresponds to the COMPFILE command parameter `SYNC READAHEAD(`*`ralimit ramatch`*`).`

Key Synchronisation

Key synchronisation is suitable where an exact match on only data in specified key segments within the records of a record pair is necessary to identify it as a synchronised record pair. The data contained in the key segments need not be unique and records containing these key fields need not have been sorted into ascending order based on data in the key segments.

If the files contain records of different record type having synchronisation key segments that are not common to all record types, then Hierarchical Key Synchronisation must be used.

Key segments in these records are collectively known as the synchronisation key and are defined when invoking the compare files facility. Segments of the synchronisation key need not necessarily be included within the compare data.

For unformatted compare, the synchronisation key segments must be specified as key field positions and lengths.

For formatted compare that involves records of different record types but with each record containing the same synchronisation key (e.g. KSDS records), the synchronisation key segments should be specified as a key field positions and lengths. In doing so, the compare files utility will ignore the applied record structure when performing key synchronisation.

Formatted compare involving only records of the same record type may specify the synchronisation key segments using the formatted record field names or field references.
Formatted compare involving records of different record types with synchronisation key segments specified using formatted record field names or field references forces hierarchical key synchronisation.

Key Synchronisation criteria requires that data must match in all key segments defined within the records comprising the current record pair.

Unsorted Key Synchronisation

Where the keyed records **are not** sorted by the synchronisation key in ascending order, unsorted key synchronisation should be used.

Unsorted key synchronisation employs a read-ahead policy to synchronise record pairs, including use of a read-ahead limit and read-ahead matching record count.

Unsorted key synchronisation criteria is similar to read-ahead synchronisation except that the areas of matching data within records comprising the current record pair, are defined by the synchronisation key as opposed to the compare data.

Unsorted Key synchronisation proceeds as described for read-ahead synchronisation above.

This synchronisation technique corresponds to the COMPFILE command parameters `SYNC UNSORTED KEY READAHEAD(`*`ralimit ramatch`*`).`

Sorted Key Synchronisation

Where the keyed records **are** sorted by the synchronisation key in ascending order, sorted key synchronisation should be used.

Sorted Key synchronisation proceeds as follows:

1. The synchronisation key segments of the current record pair are checked to determine whether synchronisation criteria are satisfied. If so, the compare data in this synchronised record pair is compared and flagged as being matched or having been changed as appropriate.

2. If not a synchronised pair, then records that follow the current record pair are read sequentially from the file containing the record with the lower synchronisation key value. Reading stops when End-of-Data is encountered or a record is found that has synchronisation key data that matches, or is greater than, the synchronisation key data of the other record in the current record pair.

3. If End-of-Data is encountered then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as being inserts or deletes as appropriate.

4. If a record with a matching or greater synchronisation key is found then this record becomes the new record in the current record pair and all records between it and the original current record are flagged as having been either inserted or deleted as appropriate.
   Note that the record in the original current record pair with the higher synchronisation key, remains in the current record pair.

   The synchronisation process is repeated for the new current record pair.

This synchronisation technique corresponds to the COMPFILE command parameters `SYNC KEY`.

Hierarchical Key Synchronisation

Hierarchical key synchronisation is performed automatically for Hierarchical Compare.

Hierarchical key synchronisation is suitable where a hierarchy exists between formatted records or record segments of different record types in the same file. e.g. A file may be arranged in a hierarchy of records or record segments detailing ORDERS, ORDER ITEMS and ITEM PARTS so that ORDERS base record segments are followed by a number of ORDER ITEMS record segments followed by a number of ITEM PARTS record segments.

Note that, if no hierarchical relationship exists between formatted records assigned different keyed record types within the same file, then a separate non-hierarchical compare files operation should be performed instead, one each for records assigned the same record type.

The record hierarchy is maintained by key fields defined to at least one of the record types. Key fields in these records are collectively known as the synchronisation key and are defined when invoking the compare files facility. Fields defined in the synchronisation key need not necessarily be included within the compare data.

The record type synchronisation key hierarchy is established by the order in which synchronisation keys are specified for each record type. The first record type to be defined a synchronisation key is attributed the highest level (level-1) entry in the hierarchy, the next key definition is attributed the level-2 entry, etc. Record types with no defined synchronisation key are equally attributed the lowest level entry in the synchronisation key hierarchy.

Data contained in the synchronisation key need not be unique and records containing a synchronisation key do not have to be sorted into ascending order based on data in the synchronisation key fields.

Hierarchical Key Synchronisation criteria requires that the following conditions are true in order identify the current record pair as a synchronised record pair:

- Keyed or unkeyed records that comprise the current record pair must be of the same record type.
- Keyed records that comprise the current record pair must have matching data in all synchronisation key fields.
- Unkeyed records that comprise the current record pair must have matching compare data and satisfy the read-ahead matching record count. (True for both sorted and unsorted hierarchical key synchronisation.)

The defining feature of both unsorted and sorted hierarchical key synchronisation processing, is that input of records from the NEW and/or OLD files stops when a keyed record is read which is rated higher in the synchronisation key hierarchy than the record being synchronised in the current record pair.
This ensures that records cannot be synchronised with records of the same record type but belonging to a different parent record type.

This rule applies equally to records comprising the current record pair. i.e. synchronisation will not be attempted for a record within the current record pair if that record is rated lower in the synchronisation key hierarchy than the other record in the current record pair.

Unsorted Hierarchical Key Synchronisation

Unsorted hierarchical key synchronisation processing is the same as for unsorted key synchronisation of formatted records but with the following additional conditions:

1. The read ahead of records from the OLD and NEW files is restricted, not only by the defined read-ahead limit (keyed records only) and End-of-Data conditions, but also by input of a record which is rated higher in the synchronisation key hierarchy than that of the record being synchronised.

2. On attempting synchronisation of a **keyed** record, the read-ahead matching record count value is ignored. This is because a matching keyed record pair may often be followed by different child record types.

3. Synchronisation of **unkeyed** records is described in *"Hierarchical Key Synchronisation of Unkeyed Records"* below.

4. If no potential synchronised record pairs are identified, then additional processing occurs as described in *"Unsynchronised Hierarchical Record Pair"* below.

This synchronisation technique corresponds to the COMPFILE command parameter `SYNC UNSORTED KEY READAHEAD(`*`ralimit ramatch`*`).`

Sorted Hierarchical Key Synchronisation

Sorted hierarchical key synchronisation may only be used if **all** keyed records are sorted in ascending order by their synchronisation key data.

The sorted hierarchical key synchronisation processing is the same as for sorted key synchronisation of formatted records but with the following additional conditions:

1. If the current record pair is comprised of records assigned record types of different levels in the synchronisation key hierarchy, then synchronisation occurs for the record with the key rated higher in the synchronisation key hierarchy.

The record with the lower rated key and all records up to the next synchronised record pair will be flagged as being inserted or deleted as appropriate.

2. Input of records from the file with the lower synchronisation key is restricted, not only by the End-of-Data condition, but also by input of a record which is rated higher in the synchronisation key hierarchy than that of the record being synchronised.
3. Synchronisation of **unkeyed** records performs read-ahead synchronisation as described in *"Hierarchical Key Synchronisation of Unkeyed Records"* below.
4. If no potential synchronised record pairs are identified, then additional processing occurs as described in *"Unsynchronised Hierarchical Record Pair"* below.

   This synchronisation technique corresponds to the COMPFILE command parameters `SYNC KEY`.

Hierarchical Key Synchronisation of Unkeyed Records

Hierarchical compare (both sorted or unsorted key synchronisation) supports records tha are assigned a record type defined without a synchronisation key. For key synchronisation purposes, these unkeyed records are considered to be lower in the synchronisation key hierarchy than any keyed record.

Synchronisation processing for unkeyed record pairs is identical for both sorted and unsorted hierarchical key synchronisation and will only be attempted if both records of the current record pair are unkeyed. This is because synchronisation of an unkeyed record is not attempted if the other record in the current record pair is keyed.

Whether or not records comprising the unkeyed record pair is assigned the same record type, standard read-ahead synchronisation processing is performed with the following exceptions:

1. The read-ahead limit is ignored for both sorted and unsorted hierarchical synchronisation. The records read will be limited by the End-of-Data condition and input of a keyed record.
2. Unlike standard sorted key synchronisation, sorted hierarchical key synchronisation supports a read-ahead matching record count in order to comply with read-ahead synchronisation of unkeyed records.

Unsynchronised Hierarchical Record Pair

For **sorted** hierarchical key synchronisation only, encountering End-of-Data before a synchronised record pair can be established will simply flag the records in the current record pair up to the End-of-Data for both files, as having been inserted or deleted as appropriate.

If, for any other reason, sorted or unsorted hierarchical key synchronisation identifies no potential synchronised record pairings for records in the current record pair, then different processing occurs depending on the records that comprise the current record pair:

Unkeyed Record Pair

If both records are unkeyed then this constitutes a record mismatch corresponding to a NEW file record insert and an OLD file record delete. The compare operation continues at the next record pair read sequentially from both files.

Keyed Record - Record Type Different

If at least one of the records is keyed and the records are of **different** record types, then processing proceeds as follows:

1. Records are read from the **same** file as the record assigned a record type with the lower level synchronisation key until one of the following is encountered.
   - ♦ The End-of-Data condition.
   - ♦ For **unsorted** hierarchical key synchronisation only, the read-ahead limit.
   - ♦ A record of the same record type or one which is rated higher in the synchronisation key hierarchy.
2. If End-of-Data is encountered then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as having been inserted or deleted as appropriate.
3. For **unsorted** hierarchical key synchronisation only, if the read-ahead limit is reached then the compare file operation terminates with with error ZZSD410E.
4. If a record of the same record type or one with a higher level synchronisation key is found then this record becomes the new record in the current record pair. All records between this record and the original current record are flagged is having been either inserted or deleted as appropriate.
   Note that the record in the current record pair with the higher level key, remains in the current record pair.

   If the new current record pair is not a synchronised record pair, then hierarchical key syncronisation processing occurs.

Keyed Record - Record Type Same

If the records are of the **same keyed** record type, then processing proceeds slightly differently for sorted and unsorted hierarchical key synchronisation.

For **unsorted** hierarchical key synchronisation:

1. Records are read from **both** files until either End-of-Data or the read-ahead limit is encountered, or a record of the same record type or one with a higher level synchronisation key is read.
2. If End-of-Data is encountered in **either** of the files, then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as having been inserted or deleted as appropriate.
3. If the read-ahead limit is reached then the compare file operation terminates with with error ZZSD410E.
4. If a record of the same record type or one with a higher level synchronisation key is found in both files then these records become the new current record pair. All records between these records and the records in the original current record pair are flagged is having been either inserted or deleted as appropriate.

   If the new current record pair is not a synchronised record pair, then unsorted hierarchical key syncronisation processing occurs.

For **sorted** hierarchical key synchronisation:

1. Records are read from the file with the higher level synchronisation key until either End-of-Data is encountered or a record is read that is assigned a record type with a higher level synchronisation key.
2. If End-of-Data is encountered, then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as having been inserted or deleted as appropriate.
3. If a record with a higher level synchronisation key is found then this record becomes the new record in the current record pair. All records between this record and the original record in the current record pair are flagged is having been either inserted or deleted as appropriate.
   Note that the record in the current record pair with the lower level synchronisation key, remains a record of the current record pair.

   The new current record pair is not synchronised record pair so sorted key syncronisation processing occurs.

Compare Files Panels

The Compare Files utility panel views (ZZSGCFO0) and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

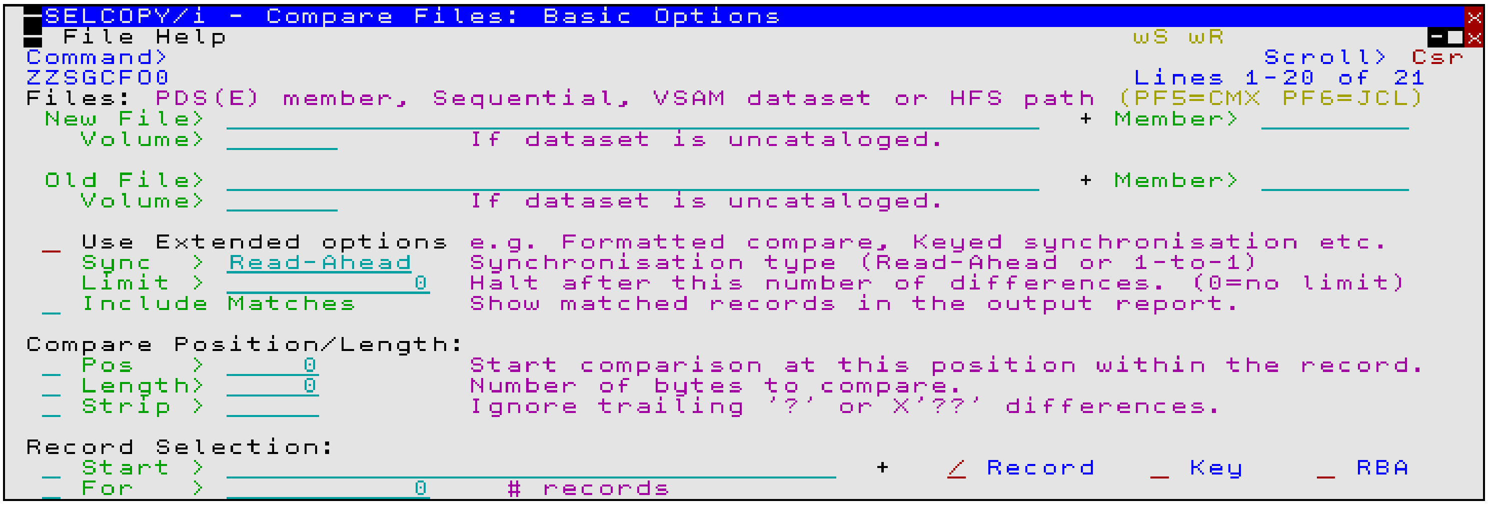
- Select 'Compare Files' from the Utilities menu.
- Execute the command COMPFile with no parameters from the command line of any window.
- Execute the prefix command "**CF**" from a file List type window. The resulting Compare Files panel window will treat the corresponding list entry as the NEW file.

By default, field entries are populated with arguments and options that were entered the last time the Compare Files Utility panels were used.

Most field entries are optional and need to be activated by entering "/" in the preceding field.

Basic Unformatted Compare Panel

Compare Files: Basic Options



```
 SELCOPY/i - Compare Files: Basic Options
 File Help                                                      wS wR
Command>                                                         Scroll> Csr
ZZSGCF00                                                     Lines 1-20 of 21
Files: PDS(E) member, Sequential, VSAM dataset or HFS path (PF5=CMX PF6=JCL)
 New File> ____________________________________________ + Member> ________
   Volume> ______         If dataset is uncataloged.

 Old File> ____________________________________________ + Member> ________
   Volume> ______         If dataset is uncataloged.

_  Use Extended options  e.g. Formatted compare, Keyed synchronisation etc.
   Sync   > Read-Ahead    Synchronisation type (Read-Ahead or 1-to-1)
   Limit  > __________0   Halt after this number of differences. (0=no limit)
_  Include Matches        Show matched records in the output report.

Compare Position/Length:
 _ Pos    > _____0        Start comparison at this position within the record.
 _ Length> _____0         Number of bytes to compare.
 _ Strip > _____          Ignore trailing '?' or X'??' differences.

Record Selection:
 _ Start > _________________________________  +  / Record   _ Key      _ RBA
 _ For   > __________0    # records
```

Figure 33. SELCOPY/i - Compare Files: Basic Options.

The **Compare Files: Basic Options** panel view is the first displayed when the Compare Files utility is started interactively.

It is anticipated that most file compare requirements will be satisfied by this panel view. Therefore, having typed entries in the required panel fields, simply pressing the <Enter> key or, if configured, double-clicking the left mouse button will will action the file compare in the foreground.

Alternatively, the user can:

- Press **PF5** to generate COMPFILE command line syntax.
- Press **PF6** to generate a COMPFILE batch job.

If this panel view does not satisfy the user's compare file requisites, then the Extended Options option field should be selected.

Features of basic unformatted compare are described in *"Basic Unformatted Compare"*.

New/Old File>
Member>
Volume>

Identifies the NEW and OLD files to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

Use Extended Options

Select this option if Extended Unformatted Compare , Formatted Compare or Hierarchical Compare is required.

If selected then the **Compare Type - Formatted/Unformatted ?** panel view is displayed which is a springboard into the Extended Unformatted Compare Panels or Formatted Compare Panels.

Apart from the file names which are passed to the extended compare panels, all options an field values entered in the **Compare Files: Basic Options** panel view will be ignored. The user will be presented with a sequence of panels that allow specification of the extended compare options.

Extended Options should be selected only if one or more of the following are required:

◊ **Compare Position/Length** needs to be specfied separately for the NEW and OLD files e.g. compare positions 1-10 from the NEW file with positions 101-110 from the OLD file.

◊ **Record Selection** needs to be specified separately for the NEW and OLD files e.g. compare the first 100 records from the NEW file with the second 100 records from the OLD file.

◊ A record synchronisation technique is required other than 1-TO-1 or Read-Ahead with the default limit value of 100, default matching record count value of 1 and no synchronisation on blank records. e.g. Read-Ahead with a limit of 20 and matching record count 2, Unsorted Key or Sorted Key synchronisation.
◊ Formatted Compare or Hierarchical Compare is required (both using a structure (SDO) or copy book to map record data).
◊ Records flagged as having been changed, inserted and/or deleted are to be excluded from the output report.
◊ A specific report DSN and/or output files for NEW/OLD matched/inserted/deleted/changed records are required.

Sync>

Select either **Read-Ahead** or **1-to-1**.

The **Compare Files: Basic Options** panel only offers these record synchronisation techniques. For a greater choice (including Unsorted Key and Sorted Key synchronisation), select the Use Extended Options option.

Read-Ahead uses default read ahead limit value of 100 records, matching record count of 1 record and bypasses synchronisation on blank records.

This field corresponds to COMPFILE parameter SYNC.

`Limit>`

Use this option in order terminate the compare process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file (or record selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

`Include Matches`

Corresponding records from both NEW and OLD files that match are to be included as matched records in the output report file.

This field corresponds to the COMPFILE parameter INCMATCHED.

`Pos>`

Record data from both the NEW and OLD files will be compared from this position onwards, for the specified length, or to the end of the record if no length is specified.

This field corresponds to the COMPFILE parameter STARTCOL.
Default is 1.

`Length>`

Record data from both the NEW and OLD files will be compared for this length.

This field corresponds to the COMPFILE parameter COMPARELEN.
Default is the length from the start position, specified in the **Pos>** field, to the end of the record.

`Strip>`

Record data from both the NEW and OLD files will have any trailing occurrences of the specified character stripped before comparison is actioned.

This option is particularly useful when **fixed length records** of different length are being compared, or when fixed length records are being compared with **variable length records**.

The single character may be specified as a literal **x** or **'x'**, which will be upper cased before stripping occurs, character string **C'x'** (character case preserved) or a hexadecimal string **X'nn'**.

This field corresponds to the COMPFILE parameter STRIP.
Default is blank (X'40').

`Start>`

Defines the start record for comparison in both the NEW and OLD files.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters STARTREC, STARTRBA and STARTKEY.
There is no default.

`Record | Key | RBA`

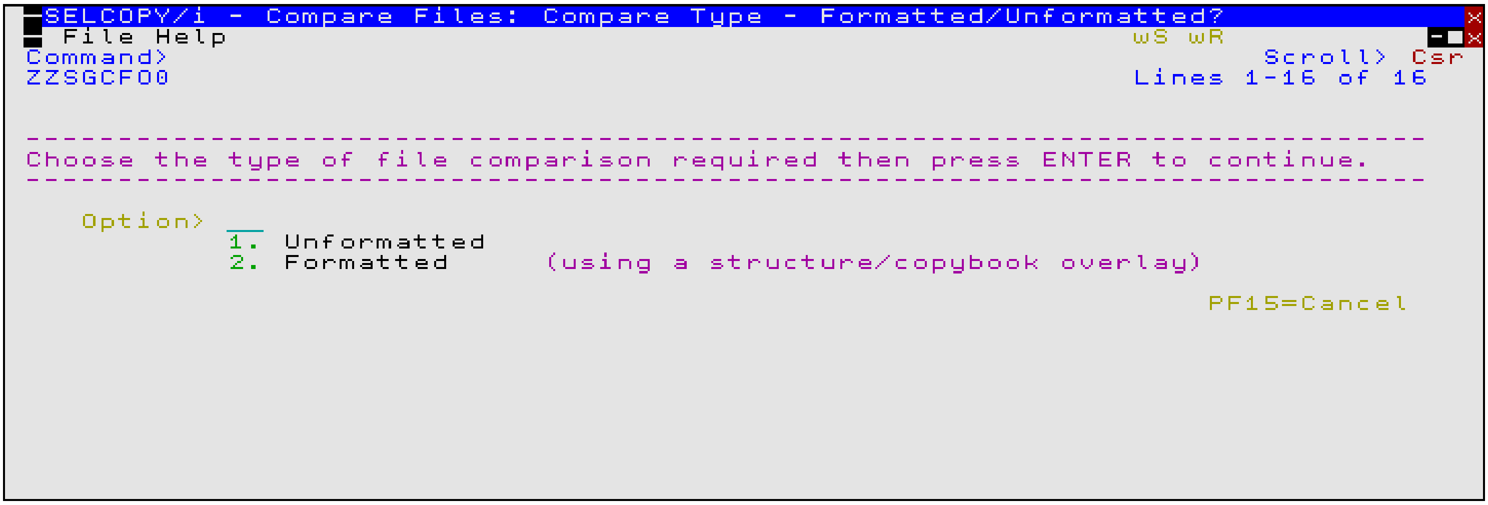
Identifies the type of start value as described by Start above.

`For>`

Specifies the maximum number of records to be compare from both files.

This field corresponds to the COMPFILE parameter FOR.
Default is 0 (all records).

Compare Type - Formatted/Unformatted ?



```
SELCOPY/i - Compare Files: Compare Type - Formatted/Unformatted?
 File Help                                                   wS wR
Command>                                                            Scroll> Csr
ZZSGCF00                                                      Lines 1-16 of 16

----------------------------------------------------------------------------
Choose the type of file comparison required then press ENTER to continue.
----------------------------------------------------------------------------

   Option> __
           1. Unformatted
           2. Formatted      (using a structure/copybook overlay)

                                                                PF15=Cancel
```

Figure 34. SELCOPY/i - Compare Files: Compare Type - Formatted/Unformatted?

`Option>`

Enter the number corresponding to the type of file compare required. Alternatively, position the cursor on the required type and press the <Enter> key or, if configured, double-click the left mouse button.

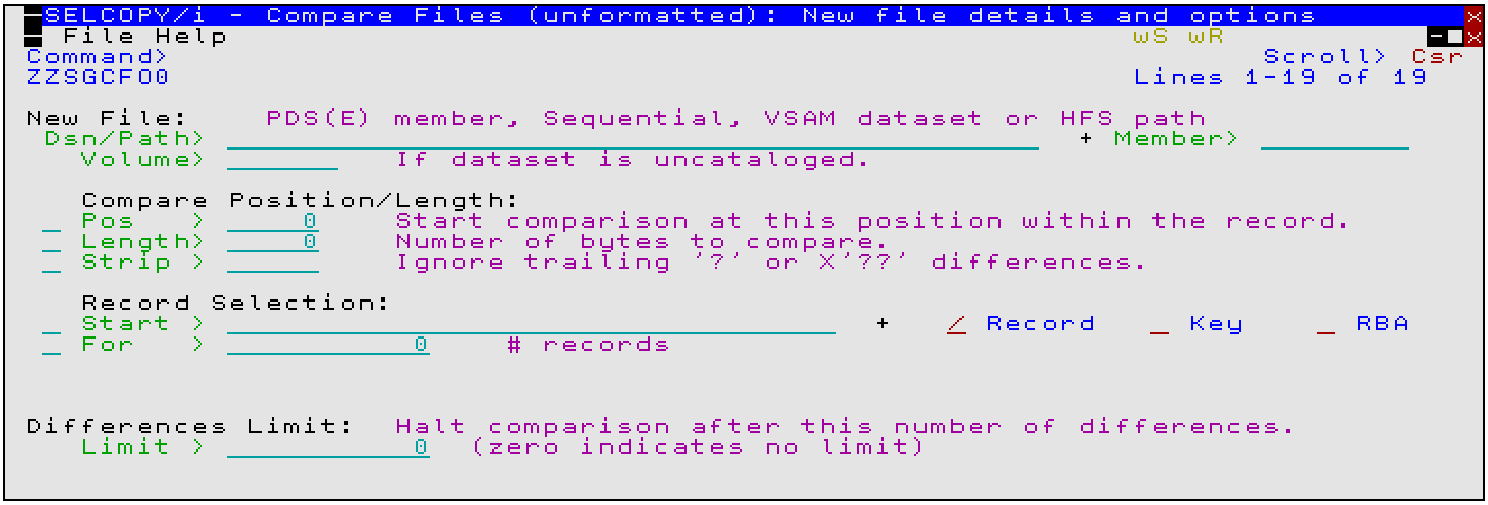
1. Unformatted
   Select this option for a extended unformatted compare of record data. Subsequent panels will offer only options suitable for extended unformatted compare.

2. Formatted
   Select this option for a formatted compare of structured records. Subsequent panels will require you to enter a structure (SDO) or COBOL/PL1 copybook/ADATA data set name.

   This option should also be selected if Hierarchical Compare is required.

Extended Unformatted Compare Panels

Compare Files (unformatted): New file details and options



```
SELCOPY/i - Compare Files (unformatted): New file details and options
 File Help                                                   wS wR
Command>                                                            Scroll> Csr
ZZSGCF00                                                      Lines 1-19 of 19

New File:    PDS(E) member, Sequential, VSAM dataset or HFS path
 Dsn/Path> ____________________________________________ + Member> ________
   Volume> ______    If dataset is uncataloged.

   Compare Position/Length:
 _ Pos    > _____0    Start comparison at this position within the record.
 _ Length> _____0    Number of bytes to compare.
 _ Strip > _____     Ignore trailing '?' or X'??' differences.

   Record Selection:
 _ Start > _________________________________  +  / Record   _ Key    _ RBA
 _ For   > __________0      # records


Differences Limit:  Halt comparison after this number of differences.
   Limit > __________0  (zero indicates no limit)
```

Figure 35. SELCOPY/i - Compare Files (unformatted): New file details and options.

The **Compare Files (unformatted): New file details and options** panel view is the first displayed for Extended Unformatted Compare, following selection of "Unformatted" from the Compare Type - Formatted/Unformatted ? panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Old file details and options.

Differences between Basic and Extended unformatted compare are described in *"Extended Unformatted Compare"*.

Dsn/Path>
Member>
Volume>

Identifies the NEW file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

Pos>

Record data from the NEW file will be compared from this position onwards, for the specified length, or to the end of the record if no length is specified.

This field corresponds to the COMPFILE parameter NSTARTCOL.
Default is 1.

Length>

Record data from the NEW file will be compared for this length.

If the compared record data in the NEW and OLD files are of different lengths, then a mismatch is inevitable and the record will be flagged as having been changed, inserted or deleted as appropriate to the employed synchronisation technique.

This field corresponds to the COMPFILE parameter NCOMPARELEN.
Default is the length from the start position, specified in the **Pos>** field, to the end of the record.

Strip>

Record data from the NEW file will have any trailing occurrences of the specified character stripped before comparison is actioned.

This option is particularly useful when **fixed length records** of different length are being compared, or when fixed length records are being compared with **variable length records**.

The single character may be specified as a literal **x** or **'x'**, which will be upper cased before stripping occurs, character string **C'x'** (character case preserved) or a hexadecimal string **X'nn'**.

This field corresponds to the COMPFILE parameter NSTRIP.
Default is blank (X'40').

Start>

Defines the record in the NEW file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters NSTARTREC, NSTARTRBA and NSTARTKEY.
There is no default.

Record | Key | RBA

Identifies the type of start value as described by Start above.

For>

Specifies the maximum number of records to be compared from the NEW file. The compare operation stops if this threshold is encountered even if the equivalent threshold for OLD file records has not been reached.

This field corresponds to the COMPFILE parameter NFOR.
Default is 0 (all records).

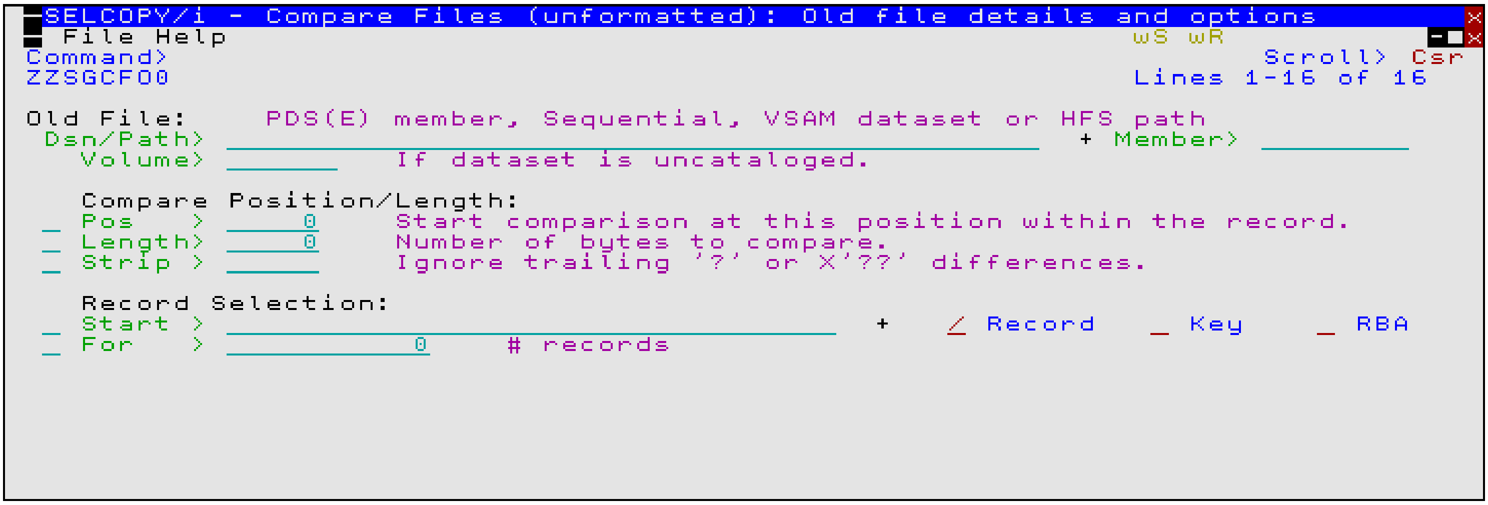
Limit>

Use this option in order terminate the compare process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file (or record selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

Compare Files (unformatted): Old file details and options



```
SELCOPY/i - Compare Files (unformatted): Old file details and options
 File Help                                                      wS wR
Command>                                                         Scroll> Csr
ZZSGCF00                                                     Lines 1-16 of 16

Old File:     PDS(E) member, Sequential, VSAM dataset or HFS path
 Dsn/Path> ____________________________________________  + Member> ________
   Volume> ______      If dataset is uncataloged.

   Compare Position/Length:
 _ Pos    > _____0     Start comparison at this position within the record.
 _ Length > _____0     Number of bytes to compare.
 _ Strip  > _____      Ignore trailing '?' or X'??' differences.

   Record Selection:
 _ Start  > _________________________________   +   / Record   _ Key    _ RBA
 _ For    > ___________0     # records
```

Figure 36. SELCOPY/i - Compare Files (unformatted): Old file details and options.

The **Compare Files (unformatted): Old file details and options** panel view is displayed following the Compare Files (unformatted): New file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Re-synchronisation options.

Dsn/Path>
Member>
Volume>

Identifies the OLD file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

Pos>

Record data from the OLD file will be compared from this position onwards, for the specified length, or to the end of the record if no length is specified.

This field corresponds to the COMPFILE parameter OSTARTCOL.
Default is 1.

Length>

Record data from the OLD file will be compared for this length.

If the compared record data in the NEW and OLD files are of different lengths, then a mismatch is inevitable and the record will be flagged as having been changed, inserted or deleted as appropriate to the employed synchronisation technique.

This field corresponds to the COMPFILE parameter OCOMPARELEN.
Default is the length from the start position, specified in the **Pos>** field, to the end of the record.

Strip>

Record data from the OLD file will have any trailing occurrences of the specified character stripped before comparison is actioned.

This option is particularly useful when **fixed length records** of different length are being compared, or when fixed length records are being compared with **variable length records**.

The single character may be specified as a literal **x** or **'x'**, which will be upper cased before stripping occurs, character string **C'x'** (character case preserved) or a hexadecimal string **X'nn'**.

This field corresponds to the COMPFILE parameter OSTRIP.
Default is blank (X'40').

`Start>`

Defines the record in the OLD file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters OSTARTREC, OSTARTRBA and OSTARTKEY.
There is no default.

`Record | Key | RBA`

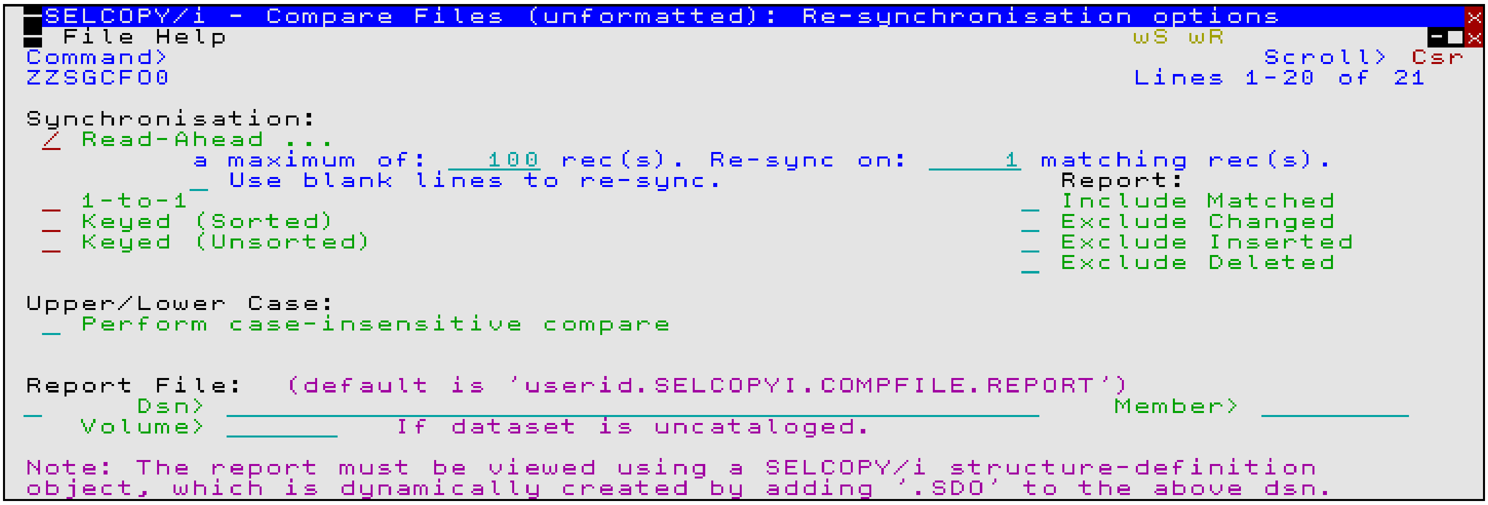
Identifies the type of start value as described by Start above.

`For>`

Specifies the maximum number of records to be compared from the OLD file. The compare operation stops if this threshold is encountered even if the equivalent threshold for NEW file records has not been reached.

This field corresponds to the COMPFILE parameter OFOR.
Default is 0 (all records).

Compare Files (unformatted): Re-synchronisation options



```
 SELCOPY/i - Compare Files (unformatted): Re-synchronisation options
  File Help                                                  wS wR
Command>                                                     Scroll> Csr
ZZSGCF00                                                 Lines 1-20 of 21

Synchronisation:
 / Read-Ahead ...
        a maximum of: ___100 rec(s). Re-sync on: _____1 matching rec(s).
        _ Use blank lines to re-sync.                       Report:
 _ 1-to-1                                                  _ Include Matched
 _ Keyed (Sorted)                                          _ Exclude Changed
 _ Keyed (Unsorted)                                        _ Exclude Inserted
                                                           _ Exclude Deleted

Upper/Lower Case:
 _ Perform case-insensitive compare

Report File:  (default is 'userid.SELCOPYI.COMPFILE.REPORT')
_      Dsn> ______________________________________       Member> ________
  Volume> ______     If dataset is uncataloged.

Note: The report must be viewed using a SELCOPY/i structure-definition
object, which is dynamically created by adding '.SDO' to the above dsn.
```

Figure 37. SELCOPY/i - Compare Files (unformatted): Re-synchronisation options.

The **Compare Files (unformatted): Re-synchronisation options** panel view is displayed following the Compare Files (unformatted): Old file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will do the following:

- If Read-Ahead or 1-to-1 synchronisation is selected, the next Extended Unformatted Compare panel view, Compare Files (unformatted): Output Files is displayed.
- If Keyed (Sorted) or Keyed (Unsorted) synchronisation is selected, then panel Compare Files (unformatted): Specify Key fields is displayed.

`Read-Ahead ...`

Select this option to use read-ahead record synchronisation.

Read-ahead synchronisation technique is suitable where the NEW and OLD files are predominantly comprise equal records, although some may have been changed, inserted or deleted.

When a record mismatch is detected, the compare files utility will attempt to resynchronsise the current, mismatching records by reading a specified number of records, first from the OLD file then from the NEW file, in order to find a match on a specified number of consecutive records. If successful, a synchronised record pair may be established and the compare operation continued from these records.

Records that have been skipped as a result of the read-ahead synchronisation are flagged as having been inserted or deleted as appropriate.

For a detailed description, see *"Read-Ahead Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC READAHEAD.

`a maximum of `*`RALimit`*` rec(s).`

The maximum number of records to read-ahead in each file when attempting to establish a synchronised record pair.

For efficiency, this value should be only one more than the maximum number of expected consecutive non-matching record pairs.

This field corresponds to the number *n1* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*).
Default value is 100.

`Re-sync on `*`RAMatch`*` matching rec(s).`

The number of consecutive matching record pairs that are required in order to establish a synchronised record pair. If satisfied, the first matching record pair is identified as a synchronised record pair.

For text files such as program source, where blank comment lines are common-place, then a match on a single line is likely to produce a false synchronised record pair that results in a less accurate report. In these circumstances a higher *RAMatch* value is advised.

This field corresponds to the number *n2* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*).
Default value is 1.

`Use blank lines to re-sync.`

Since Read-ahead synchronisation is most commonly used on text type files, where matches on blank records are likely to produce false synchronised record pairs, then, by default, blank lines are ignored when encountered as part of the read-ahead synchronisation process. i.e. a matching blank line will require a further match on the next consecutive, non-blank record pair in order to qualify.

Select this option if you wish to bypass this feature.

This field corresponds to the COMPFILE READAHEAD parameter SYNCONBLANK.

`1-to-1`

Select this option to use 1-TO-1 record synchronisation.

For 1-TO-1 synchronisation the files are assumed to contain corresponding records, so no attempt is made to resynchronise.

For a detailed description, see *"1-TO-1 Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC 1TO1.

`Keyed (Sorted)`

Select this option to use Sorted Key Synchronisation.

Keyed (Sorted) synchronisation type is suitable where the NEW and OLD files are sorted in ascending order based on one or more key segments within each record.

If this option is selected then a sub-panel will be opened, prompting the user to specify the required key segment(s).

An OLD and NEW file record may then be identified as a synchronised record pair when there is an exact match in all key segments of the record.

Where data mismatches occur in other parts of the records comprising the synchronised record pair, then the record is flagged as having been **changed**.

Records that are not established as being one of a synchronised record pair are reported as having been **inserted** or **deleted** as appropriate.

Sorted key synchronisation of unformatted data occurs by reading records from the file with the lower key data until a record with matching or higher key data is read. Intervening records are then treated as having been inserted or deleted.

For a detailed description, see *"Key Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC KEY.

Keyed (Unsorted)

Select this option to use Unsorted Key Synchronisation.

Like the **Keyed (Sorted)** option, Keyed (Unsorted) synchronisation type is suitable where the NEW and OLD file records may be identified as a synchronised record pair by an exact match in all key segments of the record. However, records are not sorted into ascending order based on these key segments.

Synchronisation of keyed unsorted records occurs using the read-ahead method as described for Read-Ahead synchronisation above. The difference being that data need only match in the defined key segments to qualify as a potential synchronised record pair.

The associated read-ahead record limit, number of consecutive matching record pairs and blank line synchronisation may also be specified for unsorted key synchronisation using the same fields used for Read-Ahead synchronisation. Namely "a maximum of *RALimit* rec(s).", "Re-sync on *RAMatch* matching rec(s)." and "Use blank lines to re-sync."

This option corresponds to COMPFILE parameters SYNC UNSORTED KEY.

Include Matched

Select this option to include matching records in the output report file.

This option corresponds to COMPFILE parameter INCMATCHED.

Exclude Changed

Select this option to exclude changed records from the output report file.

This option corresponds to COMPFILE parameter EXCHANGED.

Exclude Inserted

Select this option to exclude inserted records from the output report file.

This option corresponds to COMPFILE parameter EXINSERTED.

Exclude Inserted

Select this option to exclude deleted records from the output report file.

This option corresponds to COMPFILE parameter EXDELETED.

Perform case-insensitive compare

Select this option to perform a case insensitive compare. For unformatted compare, **all data** will be translated to upper case before comparison.

This option corresponds to COMPFILE parameter CASEINSENSITIVE (synonym CASEIGNORE).

Report File:
Dsn>
Member>
Volume>

If the Report File option field is selected, then these fields identify the fileid of the file to which the compare files utility report records will be written. Dataset names must be fully qualified, quotes being unnecessary but permitted.

The report is a structured data file designed to be browsed (not printed) using an SDE structure definition object (SDO), which will also be generated by the compare files utility.

The associated SDO fileid is constructed simply by adding **.SDO** to the report fileid. Therefore, for the DSN of the report file is restricted to 40 bytes in length.
Report output to an HFS dataset is not currently supported.

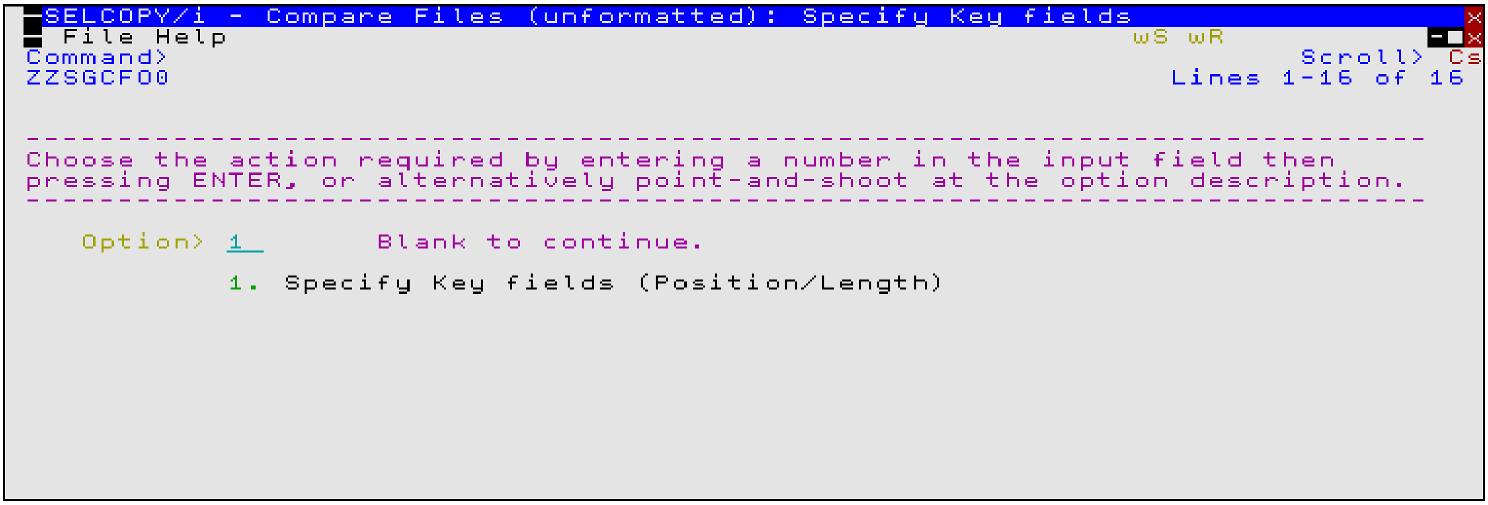
If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If this option is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

These fields correspond to COMPFILE parameter REPORT.

Compare Files (unformatted): Specify Key fields



```
SELCOPY/i - Compare Files (unformatted): Specify Key fields
 File Help                                                     wS wR
Command>                                                                Scroll> Cs
ZZSGCF00                                                           Lines 1-16 of 16

---------------------------------------------------------------------------
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
---------------------------------------------------------------------------

   Option> 1_       Blank to continue.

           1. Specify Key fields (Position/Length)
```

Figure 38. SELCOPY/i - Compare Files (unformatted): Specify Key fields.

The **Compare Files (unformatted): Specify Key fields** panel view is displayed following the Compare Files (unformatted): Re-synchronisation options panel view if Keyed (Sorted) or Keyed (Unsorted) synchronisation is selected.

`Option>`

Enter the number corresponding to the type of key field specification. Alternatively, position the cursor on the required type and press the <Enter> key or, if configured, double-click the left mouse button.

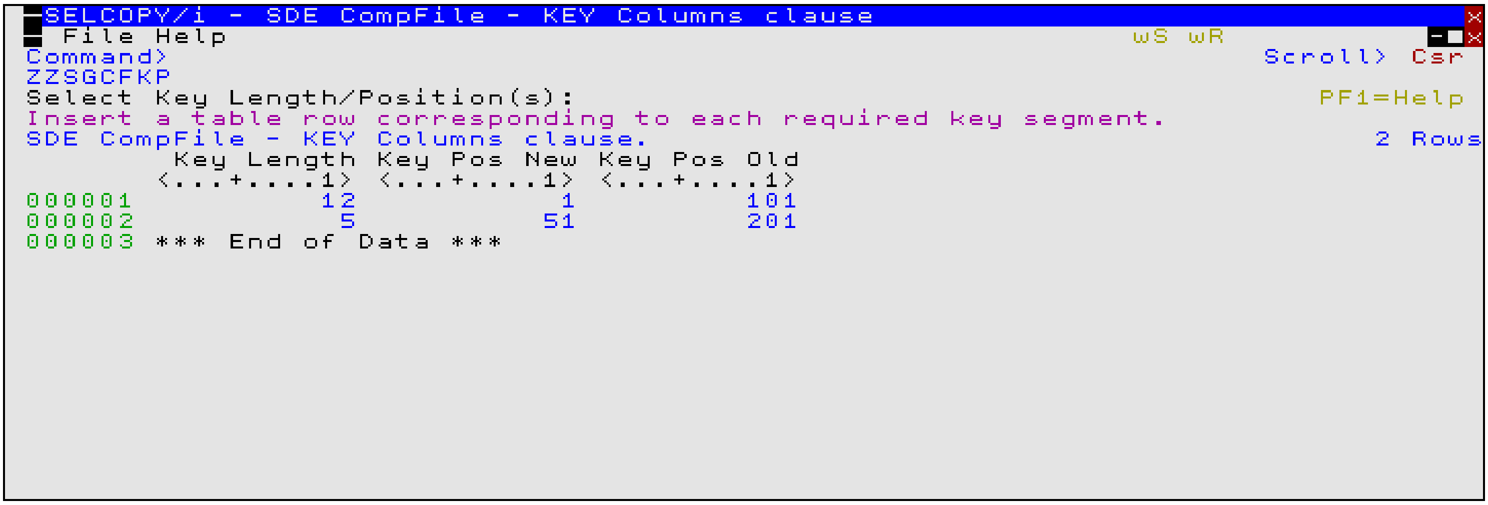
For unformatted compare, key segments may be specified by position and length only.

A blank in this field will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Output Files.

1. **Specify Key fields (Position/Length)**
   A separate panel will be displayed in which the user can enter a table row entry for each required key segment.

   Each table row entry consists of the key length, and position in both NEW and OLD file records.

Compare Files: Specify Key Pos/Len



```
SELCOPY/i - SDE CompFile - KEY Columns clause
 File Help                                                     wS wR
Command>                                                              Scroll> Csr
ZZSGCFKP
Select Key Length/Position(s):                                         PF1=Help
Insert a table row corresponding to each required key segment.
SDE CompFile - KEY Columns clause.                                       2 Rows
        Key Length Key Pos New Key Pos Old
       <...+....1> <...+....1> <...+....1>
000001          12           1         101
000002           5          51         201
000003 *** End of Data ***
```

Figure 39. SELCOPY/i - Compare Files - Specify Key Pos/Len.

The **Compare Files: Specify Key Pos/Len** panel (ZZSGCFKP) is displayed following selection of option 1. from the Compare Files (unformatted): Specify Key fields panel view.

Standard SELCOPY/i table editing techniques should be used to add a table row entry for each required key segment.

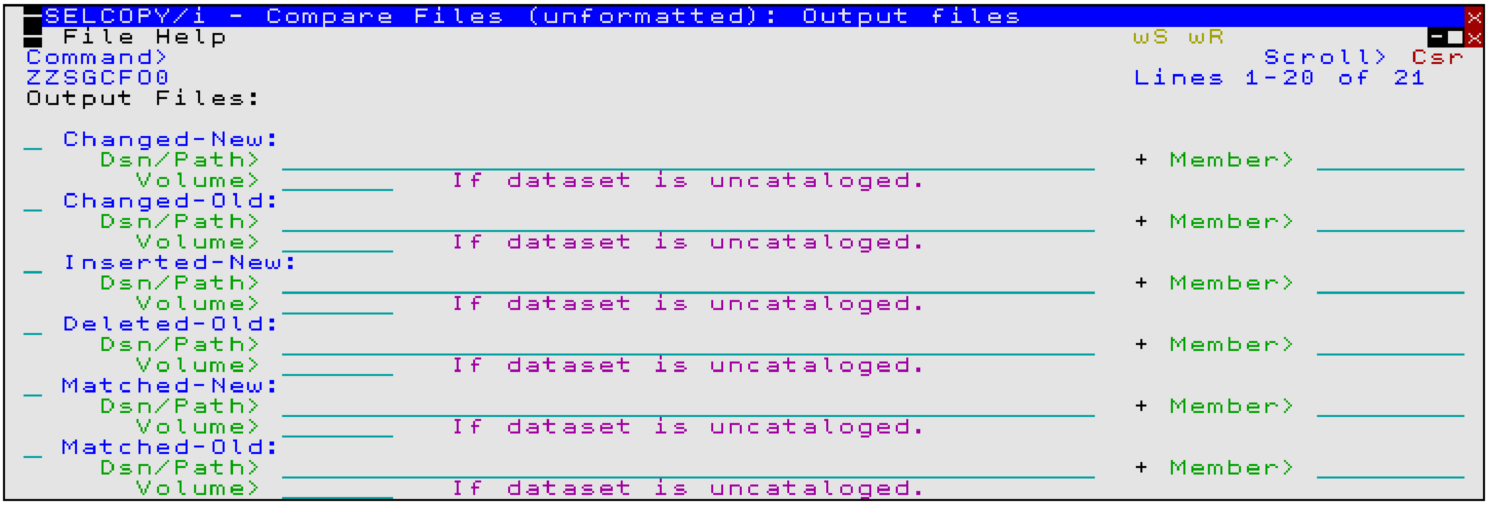
Each table row entry consists of a key length and key start positions in the NEW and OLD file records. Although the key length is fixed, the key position may differ in NEW and OLD file records.

For both unformatted compare and formatted compare, key segments specified using fixed positions and lengths apply to all record types. For record type specific keys (hierarchical compare), key segments must be selected by column name.

Key segments should be entered in the order in which they constitute the key. This is particularly important when identifying a record key to be used for sorted key synchronisation. For unsorted key synchronisation, the order in which key segments have been entered is the order in which the segments will be compared when establishing a synchronised record pair. Performance may be improved if key segments containing volatile data are specified first.

Pressing <PF3> to exit the panel, will also save the table of key segments and return to the **Compare Files (unformatted): Specify Key fields** panel view.

Compare Files (unformatted): Output Files



```
SELCOPY/i - Compare Files (unformatted): Output files
  File Help                                                     wS wR
Command>                                                          Scroll> Csr
ZZSGCF00                                                        Lines 1-20 of 21
Output Files:

_ Changed-New:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______      If dataset is uncataloged.
_ Changed-Old:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______      If dataset is uncataloged.
_ Inserted-New:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______      If dataset is uncataloged.
_ Deleted-Old:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______      If dataset is uncataloged.
_ Matched-New:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______      If dataset is uncataloged.
_ Matched-Old:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______      If dataset is uncataloged.
```

Figure 40. SELCOPY/i - Compare Files (unformatted): Output Files.

If Keyed synchronisation was specified, the **Compare Files (unformatted): Output Files** panel view is displayed following the Compare Files (unformatted): Specify Key fields panel view. Otherwise, this panel view is displayed following the Compare Files (unformatted): Re-synchronisation options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Options / Action.

The Output Files panel view identifies the output files to which a record from the NEW and/or OLD files are to be copied, based on its flagged status (matched, changed, inserted or deleted). The output fileid may be an HFS file path, sequential data set or PDS/PDSE library member.

Data set names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or if a dataset is specified which is a PDS/PDSE library and the member field is left blank.

If a specified output file is non-HFS and does not already exist, then it will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The data set is allocated using DCB RECFM, LRECL and BLKSIZE geometry that best matches the NEW or OLD file as appropriate.

Changed-New:
Dsn/Path>
Member>
Volume>

If the Changed-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been changed (CN), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECN *cn_fileid*.

Changed-Old:
Dsn/Path>
Member>
Volume>

If the Changed-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been changed (CO), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECO *co_fileid*.

Inserted-New:
Dsn/Path>
Member>
Volume>

If the Inserted-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been inserted (I), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEIN *in_fileid*.

Deleted-Old:
Dsn/Path>
Member>
Volume>

If the Deleted-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been deleted (D), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEDO *do_fileid*.

Matched-New:
Dsn/Path>
Member>
Volume>

If the Matched-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as being matched, are to be copied.

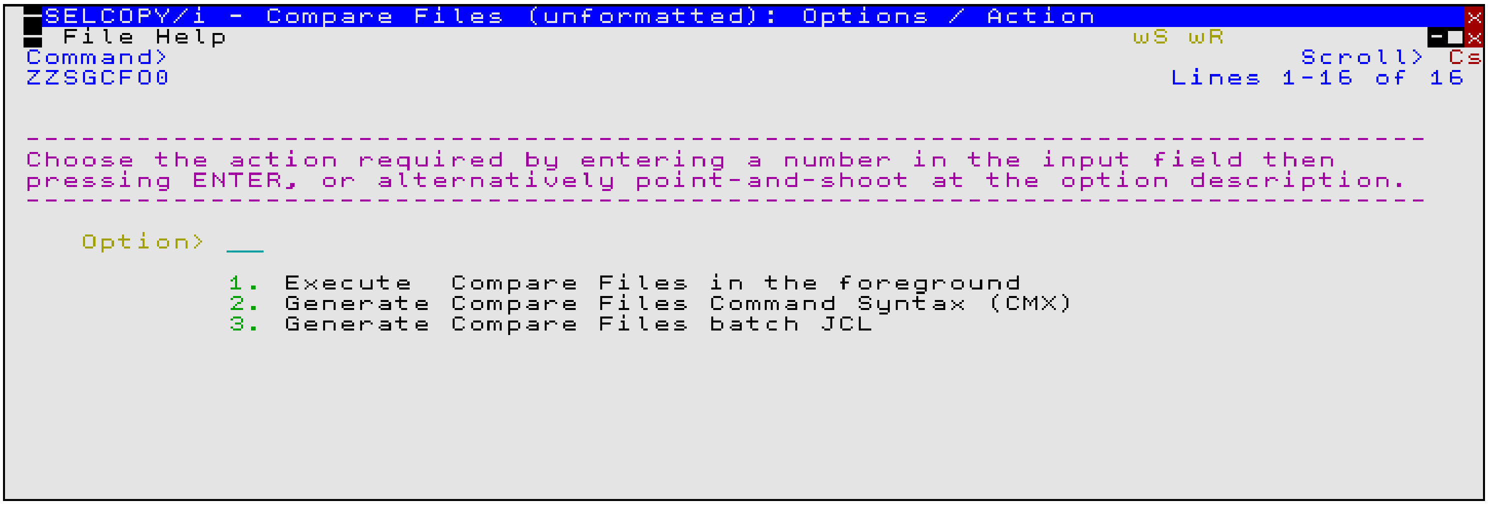
This fileid corresponds to the COMPFILE parameter WRITEMN *mn_fileid*.

Matched-Old:
Dsn/Path>
Member>
Volume>

If the Matched-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as being matched, are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEMO *mo_fileid*.

Compare Files (unformatted): Options / Action



```
SELCOPY/i - Compare Files (unformatted): Options / Action
 File Help                                                   wS wR
Command>                                                          Scroll> Cs
ZZSGCF00                                                    Lines 1-16 of 16


----------------------------------------------------------------------------
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
----------------------------------------------------------------------------

   Option> __

           1. Execute  Compare Files in the foreground
           2. Generate Compare Files Command Syntax (CMX)
           3. Generate Compare Files batch JCL
```

Figure 41. SELCOPY/i - Compare Files (unformatted): Options / Action.

The **Compare Files (unformatted): Options** / **Action** panel view is the last of the Extended Unformatted Compare panels and is displayed following the Compare Files (unformatted): Output Files panel view.

Option>

Enter the number corresponding to the action required. Alternatively, position the cursor on the action description and press the <Enter> key or, if configured, double-click the left mouse button.

1. **Execute Compare Files in the foreground**
   The compare utility will run from your SELCOPY/i session and the structured output report file will be automatically displayed in an SDE browse window view.

2. **Generate Compare Files Command Syntax (CMX)**
   COMPFILE command line syntax to run the compare files utility using the chosen options is generated and placed in a temporary CMX file. This command may be executed using CMDTEXT point-and-shoot execution <PF4> or copied into the user's HOME file and saved for future execution.

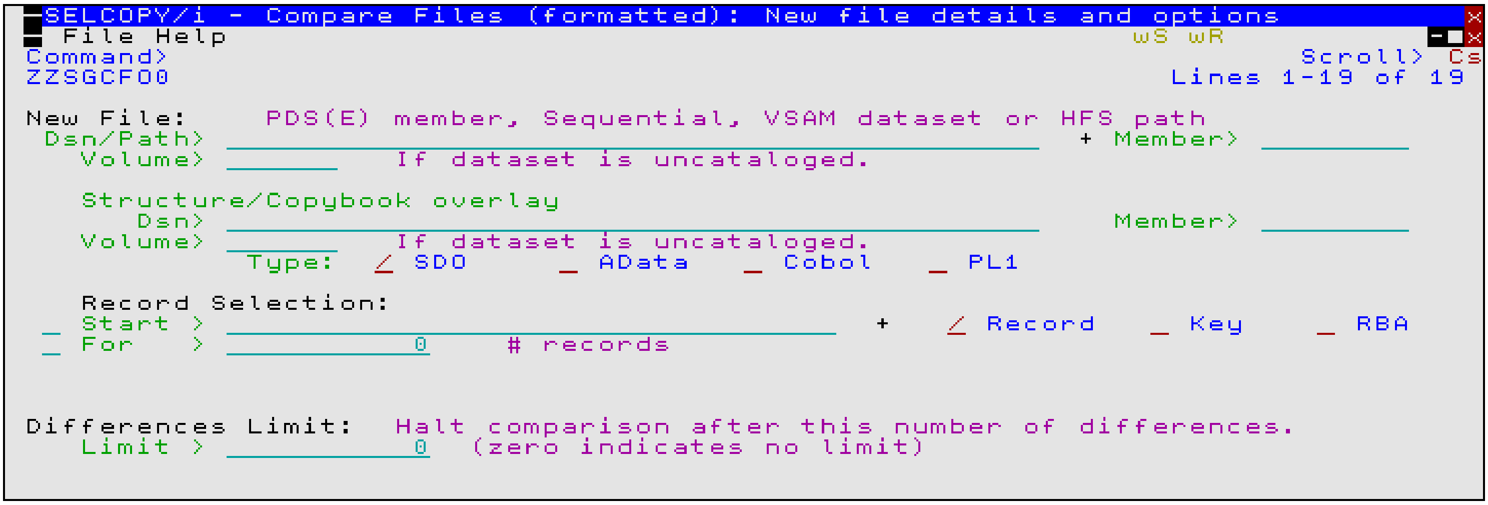
3. **Generate Compare Files batch JCL**
   Creates a JCL job stream that executes the **SDEAMAIN** program. SDEIN input comprises the COMPFILE command with parameters reflecting options specified in these panels.

   The output report, generated on execution of this batch job, may be viewed from your SELCOPY/i session by issuing the command **CFOUT *report_file_name***, or by issuing CFOUT as a prefix command against the report DSN or member name in a dataset list or library list window.

Formatted Compare Panels

The Formatted Compare panels also provide facility to specify field values necessary for Hierarchical Compare.

Compare Files (formatted): New file details and options



```
 SELCOPY/i - Compare Files (formatted): New file details and options
  File Help                                                      wS wR
Command>                                                              Scroll> Cs
ZZSGCF00                                                        Lines 1-19 of 19

New File:     PDS(E) member, Sequential, VSAM dataset or HFS path
 Dsn/Path> ____________________________________________  + Member> ________
   Volume> ______      If dataset is uncataloged.

   Structure/Copybook overlay
      Dsn> ____________________________________________    Member> ________
   Volume> ______      If dataset is uncataloged.
             Type:  / SDO      _ AData    _ Cobol    _ PL1

   Record Selection:
 _ Start > __________________________________   +   / Record   _ Key     _ RBA
 _ For   > __________0     # records


Differences Limit:  Halt comparison after this number of differences.
   Limit > __________0  (zero indicates no limit)
```

Figure 42. SELCOPY/i - Compare Files (formatted): New file details and options.

The **Compare Files (formatted): New file details and options** panel view is the first displayed for Formatted/Hierarchical Compare, following selection of "Formatted" from the Compare Type - Formatted/Unformatted ? panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Formatted Compare panel view, Compare Files (formatted): Old file details and options.

Descriptions of these types of compare may be found under *"Formatted Compare"* and *"Hierarchical Compare"*.

Dsn/Path>
Member>
Volume>

Identifies the NEW file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field is mandatory.

Structure/Copybook overlay
Dsn/Path>
Member>
Volume>
Type

Specifies the structure type (SDO, COBOL or PL1 Copybook, COBOL or PL1 ADATA output) and structure name (sequential data set or PDS/PDSE library member) to be used to map record data in NEW file.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field corresponds to the COMPFILE parameter USING SDO/COBOL/PL1/ADATA *new_structname*.
There is no default.

Start>

Defines the record in the NEW file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters NSTARTREC, NSTARTRBA and NSTARTKEY.
There is no default.

Record | Key | RBA

Identifies the type of start value as described by Start above.

For>

Specifies the maximum number of records to be compared from the NEW file. The compare operation stops if this threshold is encountered even if the equivalent threshold for OLD file records has not been reached.

This field corresponds to the COMPFILE parameter NFOR.
Default is 0 (all records).

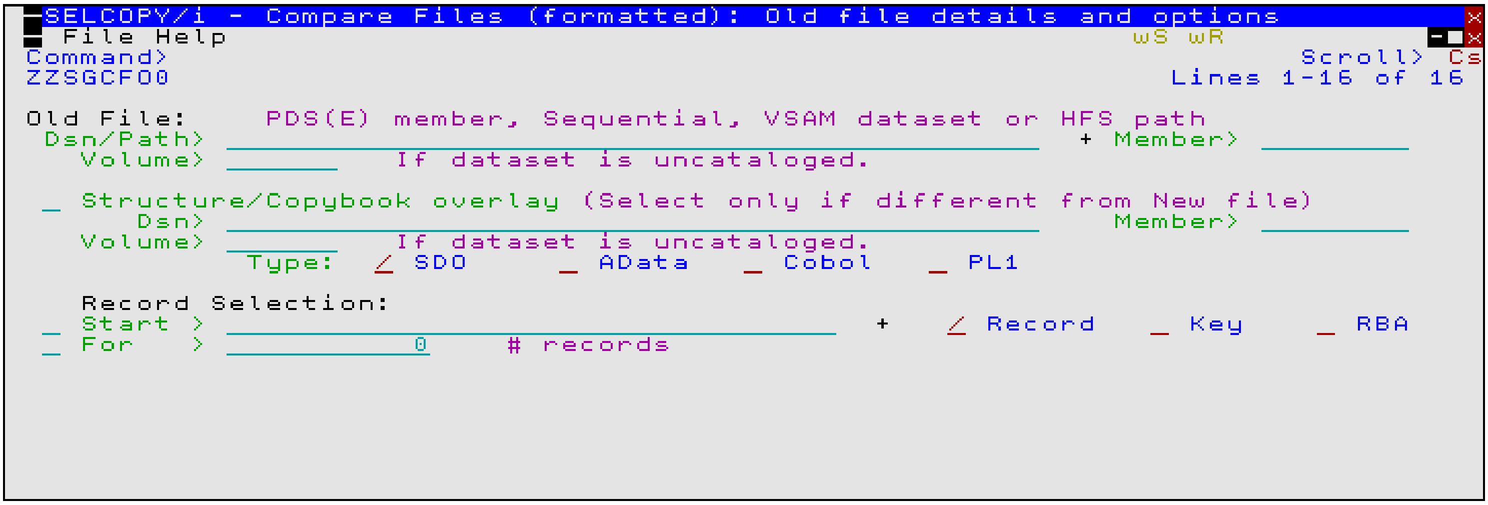
Limit>

Use this option in order terminate the compare process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file (or record selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

Compare Files (formatted): Old file details and options



```
 SELCOPY/i - Compare Files (formatted): Old file details and options
  File Help                                                  wS wR
Command>                                                         Scroll> Cs
ZZSGCF00                                                    Lines 1-16 of 16

Old File:    PDS(E) member, Sequential, VSAM dataset or HFS path
 Dsn/Path> ____________________________________________  + Member> ________
   Volume> ______    If dataset is uncataloged.

 _ Structure/Copybook overlay (Select only if different from New file)
      Dsn> ____________________________________________    Member> ________
   Volume> ______    If dataset is uncataloged.
        Type:  / SDO       _ AData     _ Cobol     _ PL1

   Record Selection:
 _ Start  > _________________________________  +   / Record    _ Key     _ RBA
 _ For    > ___________0     # records
```

Figure 43. SELCOPY/i - Compare Files (formatted): Old file details and options.

The **Compare Files (formatted): Old file details and options** panel view is displayed following the Compare Files (formatted): New file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Formatted/Hierarchical Compare panel view, Compare Files (formatted): Re-synchronisation options.

Dsn/Path>
Member>
Volume>

Identifies the OLD file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field is mandatory.

Structure/Copybook overlay
Dsn/Path>
Member>
Volume>
Type

If the Structure/Copybook overlay option field is selected, then these fields specify the structure type (SDO, COBOL or PL1 Copybook, COBOL or PL1 ADATA output) and structure name (sequential data set or PDS/PDSE library member) to be used to map record data in OLD file.

These fields are required only if a structure is to be used for the OLD file which is different to that used by the NEW file. Record type definitions of the same name must exist in both the NEW and OLD structures. Furthermore, only fields of the same name within these matching record types will be compared. i.e. Only records assigned a record type that is defined in both structures, are eligible for compare and only record data occupying fields of equal field name are compared.

Fields of the same name will be compared without error even though they may be of different data types. Note that a character field, when compared with a field of numeric data type, must contain valid numeric data (potentially including exponent 'E' or 'e', sign and/or exponent sign '+' or '-', decimal point '.' and/or commas ','.)

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field corresponds to the COMPFILE parameter USING SDO/COBOL/PL1/ADATA *old_structname*.
There is no default.

Start>

Defines the record in the OLD file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters OSTARTREC, OSTARTRBA and OSTARTKEY.
There is no default.

Record | Key | RBA

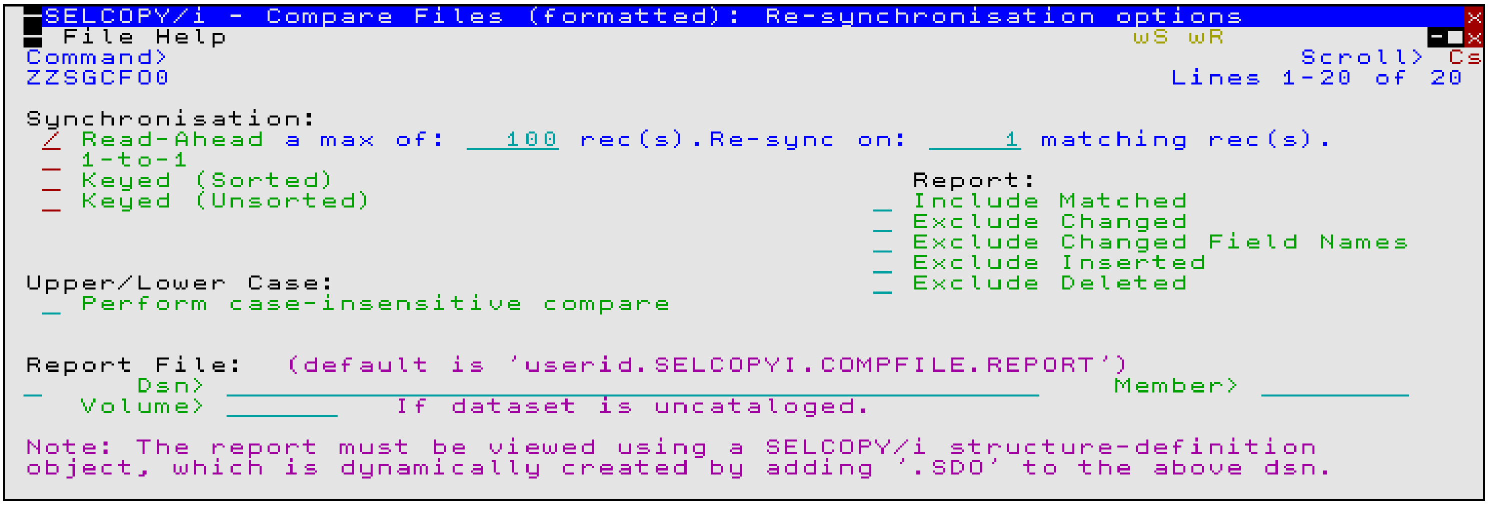
Identifies the type of start value as described by Start above.

For>

Specifies the maximum number of records to be compared from the OLD file. The compare operation stops if this threshold is encountered even if the equivalent threshold for NEW file records has not been reached.

This field corresponds to the COMPFILE parameter OFOR.
Default is 0 (all records).

Compare Files (formatted): Re-synchronisation options



```
 SELCOPY/i - Compare Files (formatted): Re-synchronisation options        x
  File Help                                                wS wR      -□x
Command>                                                       Scroll> Cs
ZZSGCF00                                                    Lines 1-20 of 20

Synchronisation:
 / Read-Ahead a max of: ___100 rec(s).Re-sync on: _____1 matching rec(s).
 _ 1-to-1
 _ Keyed (Sorted)                              Report:
 _ Keyed (Unsorted)                          _ Include Matched
                                             _ Exclude Changed
                                             _ Exclude Changed Field Names
                                             _ Exclude Inserted
Upper/Lower Case:                            _ Exclude Deleted
 _ Perform case-insensitive compare


Report File:  (default is 'userid.SELCOPYI.COMPFILE.REPORT')
_     Dsn> ____________________________________________      Member> ________
   Volume> ______    If dataset is uncataloged.

Note: The report must be viewed using a SELCOPY/i structure-definition
object, which is dynamically created by adding '.SDO' to the above dsn.
```

Figure 44. SELCOPY/i - Compare Files (formatted): Re-synchronisation options.

The **Compare Files (formatted): Re-synchronisation options** panel view is displayed following the Compare Files (formatted): Old file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will do the following:

- If Read-Ahead or 1-to-1 synchronisation is selected, the next Formatted Compare panel view, Compare Files (formatted): Output Files is displayed.
- If Keyed (Sorted) or Keyed (Unsorted) synchronisation is selected, then panel Compare Files (formatted): Specify Key fields is displayed. Note that Keyed synchronisation is required for **Hierarchical Compare**.

Read-Ahead ...

Select this option to use read-ahead record synchronisation.

Read-ahead synchronisation technique is suitable where the NEW and OLD files are predominantly comprise equal records, although some may have been changed, inserted or deleted.

When a record mismatch is detected, the compare files utility will attempt to resynchronsise the current, mismatching records by reading a specified number of records, first from the OLD file then from the NEW file, in order to find a match on a specified number of consecutive records. If successful, a synchronised record pair may be established and the compare operation continued from these records.

Note that, before attempting match data in a record pair, Read-ahead synchronisation of formatted records will first verify that the record pair are of the same record type.

Records that have been skipped as a result of the read-ahead synchronisation are flagged as having been inserted or deleted as appropriate.

For a detailed description, see *"Read-Ahead Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC READAHEAD.

a maximum of *RALimit* rec(s).

The maximum number of records to read-ahead in each file when attempting to establish a synchronised record pair.

For efficiency, this value should be only one more than the maximum number of expected consecutive non-matching record pairs.

This field corresponds to the number *n1* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 100.

Re-sync on *RAMatch* matching rec(s).

The number of consecutive matching record pairs that are required in order to establish a synchronised record pair. If satisfied, the first matching record pair is identified as a synchronised record pair.

This field corresponds to the number *n2* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 1.

1-to-1

Select this option to use 1-TO-1 record synchronisation.

For 1-TO-1 synchronisation the files are assumed to contain corresponding records, so no attempt is made to resynchronise.

For a detailed description, see *"1-TO-1 Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC 1TO1.

Keyed (Sorted)

Select this option to use Sorted Key Synchronisation or Sorted Hierarchical Key Synchronisation.

Keyed (Sorted) synchronisation type is suitable where the NEW and OLD files are sorted based on one or more key segments within each record.

If this option is selected then a sub-panel will be opened, prompting the user to specify the required key segment(s).

An OLD and NEW file record may then be identified as a synchronised record pair when there is an exact match in all key segments of the record.

Where data mismatches occur in other parts of the records comprising the synchronised record pair, then the record is flagged as having been **changed**.

Records that are not established as being one of a synchronised record pair are reported as having been **inserted** or **deleted** as appropriate.

In general, synchronisation occurs by reading records from the file with the lower key data until a record with matching or higher key data is read. Intervening records are then treated as having been inserted or deleted.

For Sorted Hierarchical Key Synchronisation, record types that have been defined a key must be sorted in ascending sort order within its hierarchical group. e.g. records of sorted keyed record type, ALBUM, has child records of sorted keyed record type, TRACKS. Records of record type TRACKS are sorted by field Track_Number (the key field) which starts at Track_Number=1 following each new ALBUM parent record.

For a detailed description, see *"Key Synchronisation"* and *"Hierarchical Key Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC KEY.

Keyed (Unsorted)

Select this option to use Unsorted Key Synchronisation and Unsorted Hierarchical Key Synchronisation.

Like the **Keyed (Sorted)** option, Keyed (Unsorted) synchronisation type is suitable where the NEW and OLD file records may be identified as a synchronised record pair by an exact match in all key segments of the record. However, records are not sorted into ascending order based on these key segments.

Synchronisation of keyed unsorted records occurs using the read-ahead method as described for Read-Ahead synchronisation above. The difference being that data need only match in the defined key segments to qualify as a potential synchronised record pair.

The associated read-ahead record limit and number of consecutive matching record pairs may also be specified for unsorted key synchronisation using the same fields used for Read-Ahead synchronisation. Namely "a maximum of *RALimit* rec(s)." and "Re-sync on *RAMatch* matching rec(s)."

This option corresponds to COMPFILE parameters SYNC UNSORTED KEY.

Include Matched

Select this option to include matching records in the output report file.

This option corresponds to COMPFILE parameter INCMATCHED.

Exclude Changed

Select this option to exclude changed records from the output report file.

For a formatted compare only, reporting a record which is flagged as having been changed will not only display the formatted record data from the NEW and OLD files, but also a number of **Field** records which identify the name of each changed field.

Note that, opting to exclude these records may result in a significant performance improvement since the process of comparing field-by-field is terminated at the first mismatch in each record.

This option corresponds to COMPFILE parameter EXCHANGED.

Exclude Inserted

Select this option to exclude inserted records from the output report file.

This option corresponds to COMPFILE parameter EXINSERTED.

Exclude Inserted

Select this option to exclude deleted records from the output report file.

This option corresponds to COMPFILE parameter EXDELETED.

Perform case-insensitive compare

Select this option to perform a case insensitive compare. For a formatted and hierarchical compare, character (AN) fields will be translated to upper case before comparison.

This option corresponds to COMPFILE parameter CASEINSENSITIVE (synonym CASEIGNORE).

Report File:
Dsn>
Member>
Volume>

If the Report File option field is selected, then these fields identify the fileid of the file to which the compare files utility report records will be written. Dataset names must be fully qualified, quotes being unnecessary but permitted.

The report is a structured data file designed to be browsed (not printed) using an SDE structure definition object (SDO), which will also be generated by the compare files utility.

The associated SDO fileid is constructed simply by adding **.SDO** to the report fileid. Therefore, for the DSN of the report file is restricted to 40 bytes in length.

Report output to an HFS dataset is not currently supported.

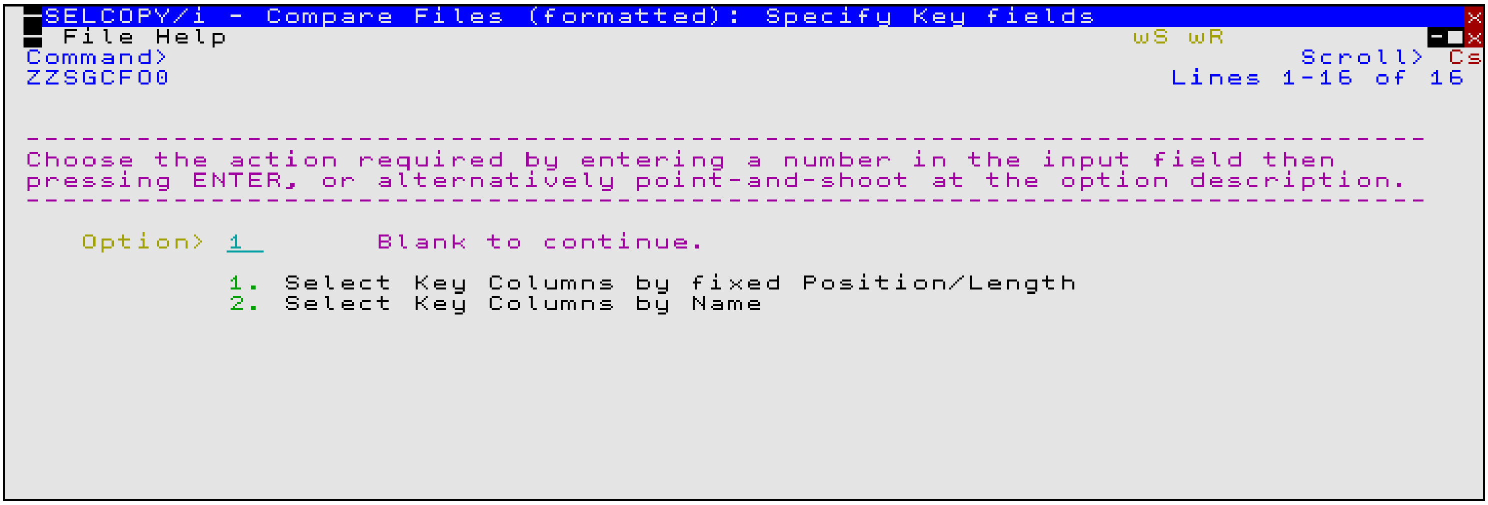
If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If this option is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

These fields correspond to COMPFILE parameter REPORT.

Compare Files (formatted): Specify Key fields



```
 SELCOPY/i - Compare Files (formatted): Specify Key fields
  File Help                                                     wS wR
Command>                                                             Scroll> Cs
ZZSGCF00                                                         Lines 1-16 of 16

-----------------------------------------------------------------------------
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
-----------------------------------------------------------------------------

   Option> 1_      Blank to continue.

           1.  Select Key Columns by fixed Position/Length
           2.  Select Key Columns by Name
```

Figure 45. SELCOPY/i - Compare Files (formatted): Specify Key fields.

Option>

Enter the number corresponding to the type of key field specification. Alternatively, position the cursor on the required type and press the <Enter> key or, if configured, double-click the left mouse button.

Formatted compare key segments may be specified by position and length or by formatted column name. Hierarchical compare key segments must be specified by formatted column name.

A blank in this field will proceed to the next Extended Formatted Compare panel view, Compare Files (formatted): Output Files.

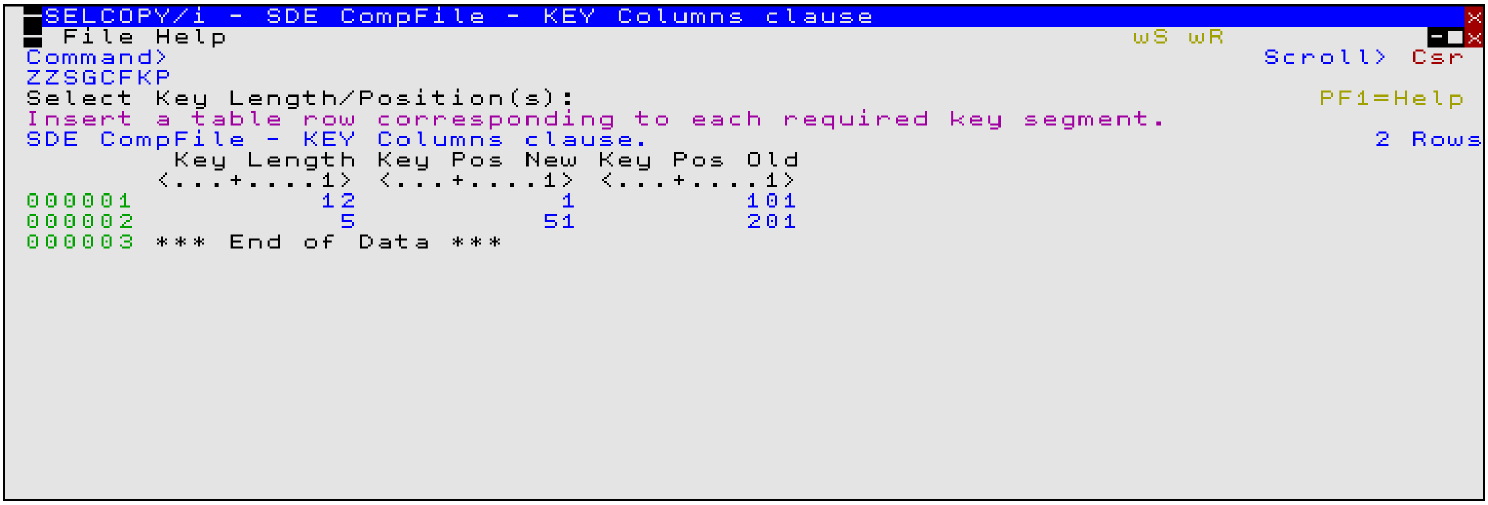
1. **Select Key Columns by fixed Position/Length**
   A separate panel will be displayed in which the user can enter a table row entry for each required key segment.

   Each table row entry consists of the key length, and position in both NEW and OLD file records.

2. **Select Key Columns by Name**
   A separate panel will be displayed containing a list of record types defined by the NEW structure (SDO) or copybook. For informational purposes only, each record type is accompanied by its **USE WHEN** condition, where specified, which enables SDE to identify each particular record type.

   From this panel, select those record types to which key segments are to be defined. A panel will be opened for each selected record type allowing the user to exclude and re-order the field columns, leaving only those to be used as key segments.

Compare Files: Specify Key Pos/Len



```
 SELCOPY/i - SDE CompFile - KEY Columns clause
 File Help                                                        wS wR
Command>                                                                 Scroll> Csr
ZZSGCFKP
Select Key Length/Position(s):                                            PF1=Help
Insert a table row corresponding to each required key segment.
SDE CompFile - KEY Columns clause.                                          2 Rows
        Key Length Key Pos New Key Pos Old
       <...+....1> <...+....1> <...+....1>
000001          12           1         101
000002           5          51         201
000003 *** End of Data ***
```

Figure 46. SELCOPY/i - Compare Files - Specify Key Pos/Len.

The **Compare Files: Specify Key Pos/Len** panel (ZZSGCFKP) is displayed following selection of option 1. from the Compare Files (formatted): Specify Key fields panel view.

Standard SELCOPY/i table editing techniques should be used to add a table row entry for each required key segment.

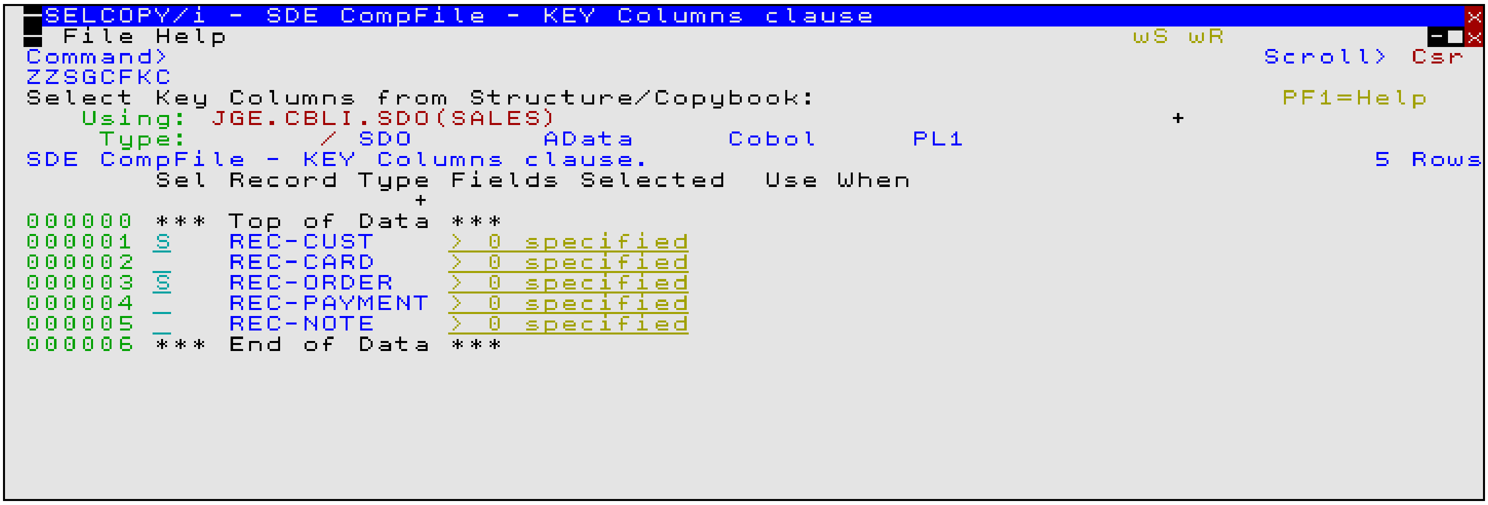
Each table row entry consists of a key length and key start positions in the NEW and OLD file records. Although the key length is fixed, the key position may differ in NEW and OLD file records.

For both unformatted compare and formatted compare, key segments specified using fixed positions and lengths apply to all record types. For record type specific keys (hierarchical compare), key segments must be selected by column name.

Key segments should be entered in the order in which they constitute the key. This is particularly important when identifying a record key to be used for sorted key synchronisation. For unsorted key synchronisation, the order in which key segments have been entered is the order in which the segments will be compared when establishing a synchronised record pair. Performance may be improved if key segments containing volatile data are specified first.

Pressing <PF3> to exit the panel, will also save the table of key segments and return to the **Compare Files (formatted): Specify Key fields** panel view.

SELCOPY/i Compare Files - KEY Columns (Record Types List)



```
 SELCOPY/i - SDE CompFile - KEY Columns clause
 File Help                                                        wS wR
Command>                                                                 Scroll> Csr
ZZSGCFKC
Select Key Columns from Structure/Copybook:                             PF1=Help
   Using: JGE.CBLI.SDO(SALES)                                   +
    Type:        / SDO      AData     Cobol     PL1
SDE CompFile - KEY Columns clause.                                          5 Rows
       Sel Record Type Fields Selected  Use When
                         +
000000 *** Top of Data ***
000001 S   REC-CUST    > 0 specified
000002 _   REC-CARD    > 0 specified
000003 S   REC-ORDER   > 0 specified
000004 _   REC-PAYMENT > 0 specified
000005 _   REC-NOTE    > 0 specified
000006 *** End of Data ***
```

Figure 47. SELCOPY/i - Compare Files - KEY Columns (Record-Types List)

The **Compare Files - Key Columns (Record Types List)** panel (ZZSGCFKC) is displayed following selection of option 2. from the Compare Files (formatted): Specify Key fields panel view.

If more than one record type is displayed in this list (i.e. the structure contains multiple record type definitions), then selecting key segment fields for any of these record types will imply Hierarchical Compare.

Select each record type for which key columns are to be specified by entering 'S' against the record type in the **Sel** column or by positioning the cursor on the required record type then either pressing the <Enter> key or, if configured, double-click the left mouse button. To deselect the record type key field definition, remove the 'S' against its entry in the 'Sel' column.

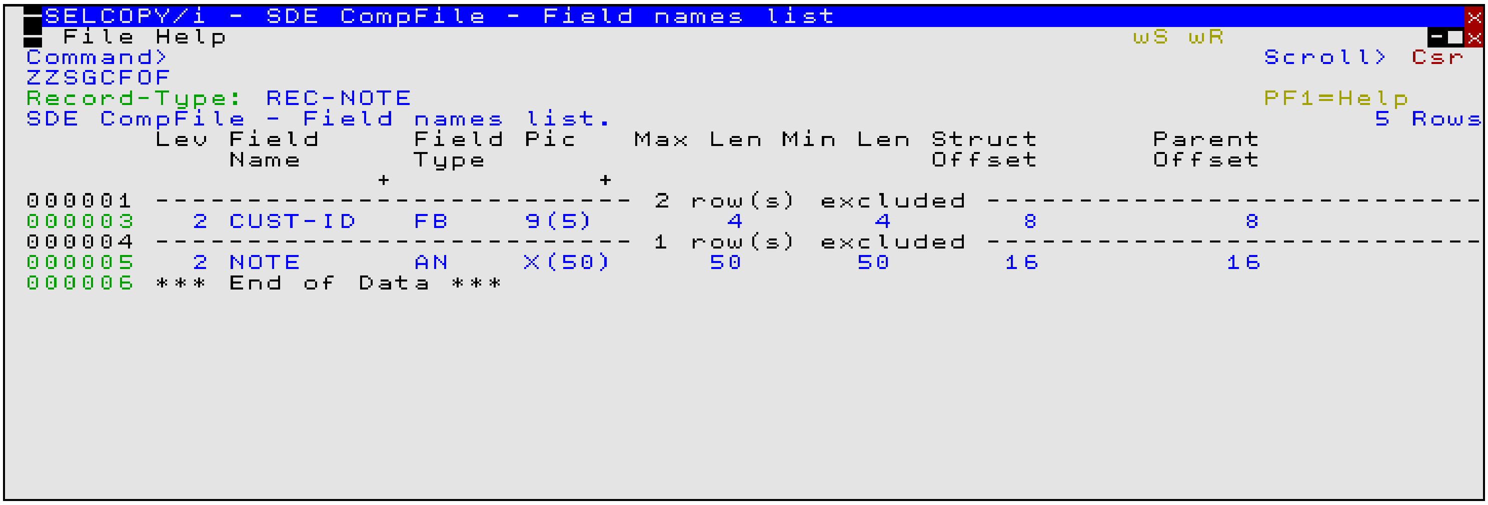
For each selected record type, the Compare Files - Select from Field Names List panel (ZZSGCFOF) is opened displaying a list of fields comprising that record type. The list of field names should be edited so that only the required key fields are diplayed in the correct order.

On return from the selectable field list, the **Fields Selected** column will be updated to indicate the number of fields included in the key.

For Hierarchical Compare it is imperative that the list of record types are ordered so that the level 1 (highest priority) keyed record type occurs first in the list, followed by the level 2 keyed record type, etc. Use standard SELCOPY/i table editing techniques to re-order the record types as required.

Pressing <PF3> to exit the panel, will also save the table of selected keyed record types and return to the **Compare Files (formatted): Specify Key fields** panel view.

SELCOPY/i Compare Files - Select from Field Names List



```
 SELCOPY/i - SDE CompFile - Field names list
  File Help                                                  wS wR
Command>                                                            Scroll> Csr
ZZSGCFOF
Record-Type: REC-NOTE                                               PF1=Help
SDE CompFile - Field names list.                                          5 Rows
       Lev Field     Field Pic     Max Len Min Len Struct      Parent
           Name      Type                          Offset      Offset
                   +           +
000001 -------------------------- 2 row(s) excluded ---------------------------
000003   2 CUST-ID   FB    9(5)      4       4       8           8
000004 -------------------------- 1 row(s) excluded ---------------------------
000005   2 NOTE      AN    X(50)    50      50      16          16
000006 *** End of Data ***
```

Figure 48. SELCOPY/i - Compare Files - Select from Field Names List.

The **Compare Files - Select from Field Names List** panel (ZZSGCFOF) is displayed for each record type selected from the Compare Files - Key Columns (Record Types List) panel.

A list of field names, defined by the selected record type, is presented to the user as an editable table. Standard SELCOPY/i table editing techniques should be used to exclude and re-order the fields so that only fields which comprise the key are displayed and in the correct order.

Only included entries are used to define the key, therefore a field name may be excluded (as opposed to deleted) in order to remove it from the key. This has the benefit that it may easily be included again later if necessary. For example, the following commands may be executed to filter (include) specific table rows:

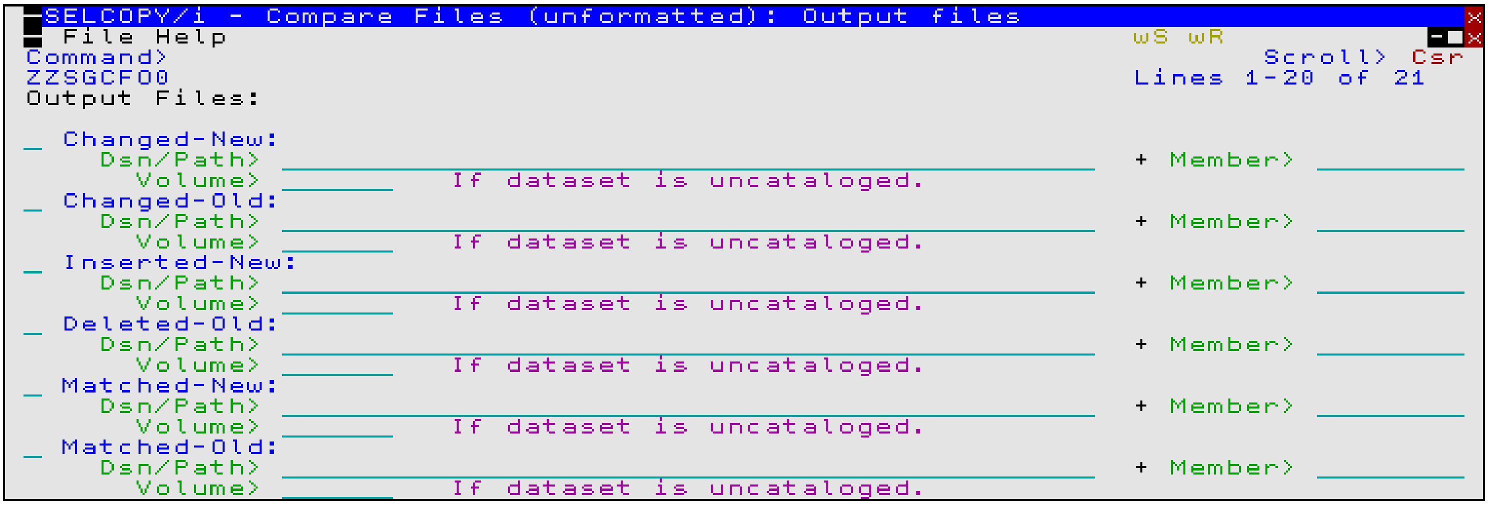
`WHERE SelectFld >> 'ABC-' or #1 << 'DEF-'`
: Exclude all entries except those where the Field Name begins with literal "**ABC-**", or "**DEF-**". Note that the Field Name column is field reference number 1.

`LESS SelectTyp='AN'`
: Exclude all entries where the Field Type is **AN**. (Entries that were already excluded will be unaffected.)

Key field segments should occur in the order in which they constitute the key. This is particularly important when identifying a record key to be used for sorted key synchronisation. For unsorted key synchronisation, the order in which key fields have been entered is the order in which the key fields will be compared when establishing a synchronised record pair. Performance may be improved if key fields containing volatile data are specified first.

Pressing <PF3> to exit the panel, will also save the table of selected key field names and return to the **Compare Files - Key Columns (Record Types List)** panel.

Compare Files (formatted): Output Files



```
 SELCOPY/i - Compare Files (unformatted): Output files
 File Help                                                  wS wR
Command>                                                           Scroll> Csr
ZZSGCF00                                                        Lines 1-20 of 21
Output Files:

_ Changed-New:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______    If dataset is uncataloged.
_ Changed-Old:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______    If dataset is uncataloged.
_ Inserted-New:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______    If dataset is uncataloged.
_ Deleted-Old:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______    If dataset is uncataloged.
_ Matched-New:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______    If dataset is uncataloged.
_ Matched-Old:
    Dsn/Path> ____________________________________________  + Member> ________
      Volume> ______    If dataset is uncataloged.
```

Figure 49. SELCOPY/i - Compare Files (formatted): Output Files.

If Keyed synchronisation was specified, the **Compare Files (formatted): Output Files** panel view is displayed following the Compare Files (formatted): Specify Key fields panel view. Otherwise, this panel view is displayed following the Compare Files (formatted): Re-synchronisation options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will proceed to the next Formatted/Hierarchical Compare panel view, Compare Files (formatted): Options / Action.

The Output Files panel view identifies the output files to which a record from the NEW and/or OLD files are to be copied, based on its flagged status (matched, changed, inserted or deleted). The output fileid may be an HFS file path, sequential data set or PDS/PDSE library member.

Data set names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or if a dataset is specified which is a PDS/PDSE library and the member field is left blank.

If a specified output file is non-HFS and does not already exist, then it will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The data set is allocated using DCB RECFM, LRECL and BLKSIZE geometry that best matches the NEW or OLD file as appropriate.

Changed-New:
Dsn/Path>
Member>
Volume>

If the Changed-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been changed (CN), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECN *cn_fileid*.

Changed-Old:
Dsn/Path>
Member>
Volume>

If the Changed-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been changed (CO), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECO *co_fileid*.

Inserted-New:
Dsn/Path>
Member>
Volume>

If the Inserted-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been inserted (I), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEIN *in_fileid*.

Deleted-Old:
Dsn/Path>
Member>
Volume>

If the Deleted-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been deleted (D), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEDO *do_fileid*.

Matched-New:
Dsn/Path>
Member>
Volume>

If the Matched-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as being matched, are to be copied.

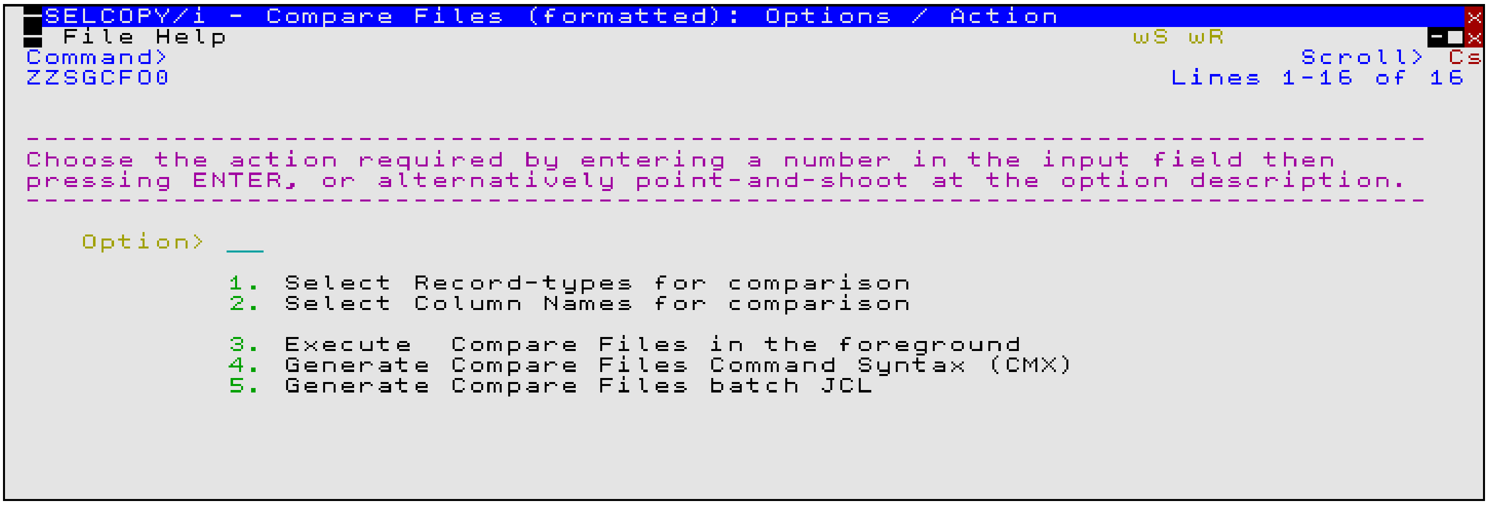
This fileid corresponds to the COMPFILE parameter WRITEMN *mn_fileid*.

Matched-Old:
Dsn/Path>
Member>
Volume>

If the Matched-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as being matched, are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEMO *mo_fileid*.

Compare Files (formatted): Options / Action



```
SELCOPY/i - Compare Files (formatted): Options / Action
 File Help                                                    wS wR
Command>                                                          Scroll> Cs
ZZSGCF00                                                     Lines 1-16 of 16

----------------------------------------------------------------------------
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
----------------------------------------------------------------------------

   Option> __

           1.  Select Record-types for comparison
           2.  Select Column Names for comparison

           3.  Execute  Compare Files in the foreground
           4.  Generate Compare Files Command Syntax (CMX)
           5.  Generate Compare Files batch JCL
```

Figure 50. SELCOPY/i - Compare Files (formatted): Options / Action.

Option>

Enter the number corresponding to the action required. Alternatively, position the cursor on the action description and press the <Enter> key or, if configured, double-click the left mouse button.

1. **Select Record Types for comparison**
   By default records of any record type are included in the compare process. Use this option to restrict the compare process to records assigned the specified record types only.

   A separate panel will be displayed containing a list of record types defined by the NEW structure (SDO) or copybook. For informational purposes only, each record type is accompanied by its **USE WHEN** condition, where specified, which enables SDE to identify each particular record type.

   From this panel, select those record types to included in the compare process.

2. **Select Column Names for comparison**
   By default all fields (from all selected record-types) are included in the compare process. Use this option to restrict the compare process to a subset of field names identified within those record types selected for compare.

   A separate panel will be displayed containing a list of record types defined by the NEW structure (SDO) or copybook. For informational purposes only, each record type is accompanied by its **USE WHEN** condition, where specified, which enables SDE to identify each particular record type.

   From this panel, select those record types for which fields are to be excluded from the compare process. Another panel will be opened for each selected record type allowing the user to exclude and field columns from the compare.

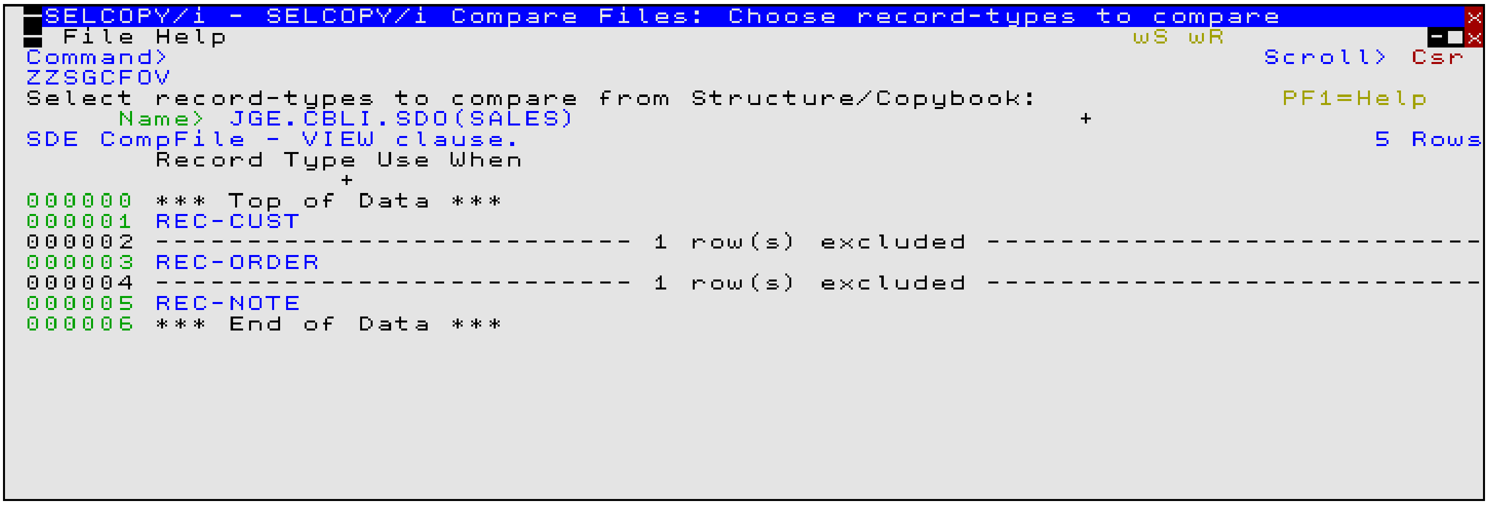
3. **Execute Compare Files in the foreground**
   The compare utility will run from your SELCOPY/i session and the structured output report file will be automatically displayed in an SDE browse window view.

4. **Generate Compare Files Command Syntax (CMX)**
   COMPFILE command line syntax to run the compare files utility using the chosen options is generated and placed in a temporary CMX file. This command may be executed using CMDTEXT point-and-shoot execution <PF4> or copied into the user's HOME file and saved for future execution.

5. **Generate Compare Files batch JCL**
   Creates a JCL job stream that executes the **SDEAMAIN** program. SDEIN input comprises the COMPFILE command with parameters reflecting options specified in these panels.

   Following The output report, generated on execution of this batch job, may be viewed from your SELCOPY/i session by issuing the command **CFOUT *report_file_name***, or by issuing CFOUT as a prefix command against the report DSN or member name in a dataset list or library list window.

SELCOPY/i Compare Files - Select Record Types to Compare



```
 SELCOPY/i - SELCOPY/i Compare Files: Choose record-types to compare
  File Help                                                          wS wR
Command>                                                              Scroll> Csr
ZZSGCFOV
Select record-types to compare from Structure/Copybook:                PF1=Help
     Name> JGE.CBLI.SDO(SALES)                                   +
SDE CompFile - VIEW clause.                                                5 Rows
       Record Type Use When
                 +
000000 *** Top of Data ***
000001 REC-CUST
000002 -------------------------- 1 row(s) excluded ---------------------------
000003 REC-ORDER
000004 -------------------------- 1 row(s) excluded ---------------------------
000005 REC-NOTE
000006 *** End of Data ***
```

Figure 51. SELCOPY/i - Compare Files - Select Record-Types.

The **Compare Files - Select Record Types to Compare** panel (ZZSGCFOV) is displayed following selection of option 1. from the Compare Files (formatted): Options / Action panel view.

This panel contains a list of all record types defined by the NEW structure (SDO) or copybook presented to the user as an editable table. Any **USE WHEN** condition, used to determine whether record data fits the record type definition, is also displayed.

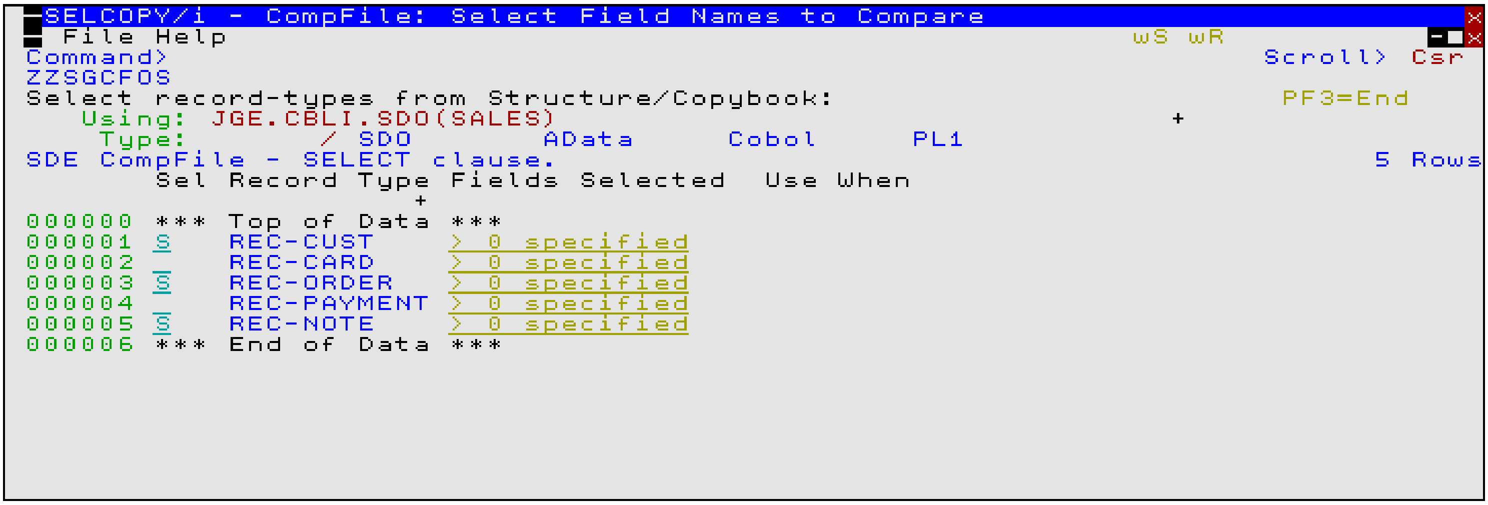
Records assigned a record type that is included in this list will be eligible for compare. Use standard SELCOPY/i table editing techniques to exclude or delete record types from this list and so exclude records assigned these record types from being compared. For example, the following command may be executed to exclude all rows except those where the record type name begins with literal "**ABC-**":

```
WHERE   ViewRT >> 'ABC-'
```

Pressing <PF3> to exit the panel, will also save the table of selected keyed record types and return to the **Compare Files (formatted): Options** / **Action** panel view.

This panel corresponds to the COMPFILE parameter VIEW *rectype*.
Default is to include all records in the NEW and OLD files in the compare regardless of whether they are assigned a record type in the NEW structure.

SELCOPY/i Compare Files - Select Field Names to Compare



```
 SELCOPY/i - CompFile: Select Field Names to Compare
  File Help                                                   wS wR
Command>                                                           Scroll> Csr
ZZSGCFOS
Select record-types from Structure/Copybook:                         PF3=End
   Using: JGE.CBLI.SDO(SALES)                                    +
    Type:       / SDO         AData      Cobol      PL1
SDE CompFile - SELECT clause.                                           5 Rows
       Sel Record Type Fields Selected  Use When
                      +
000000 *** Top of Data ***
000001 S   REC-CUST     > 0 specified
000002 _   REC-CARD     > 0 specified
000003 S   REC-ORDER    > 0 specified
000004 _   REC-PAYMENT  > 0 specified
000005 S   REC-NOTE     > 0 specified
000006 *** End of Data ***
```

Figure 52. SELCOPY/i - Compare Files - Select Field Names to Compare.

The **Compare Files - Select Field Names to Compare** panel (ZZSGCFOS) is displayed following selection of option 2. from the Compare Files (formatted): Options / Action panel view.

Select each record type for which specific field columns are to be selected by entering 'S' against the record type in the **Sel** column or by positioning the cursor on the required record type then either pressing the <Enter> key or, if configured, double-click the left mouse button. To deselect the record type key field definition, remove the 'S' against its entry in the 'Sel' column.

For each selected record type, the Compare Files - Select from Field Names List panel (ZZSGCFOF) is opened displaying a list of fields comprising that record type. The list of field names should be edited so that only the required key fields are diplayed. The order in which these fields occur in this list will be the order in which they are compared.

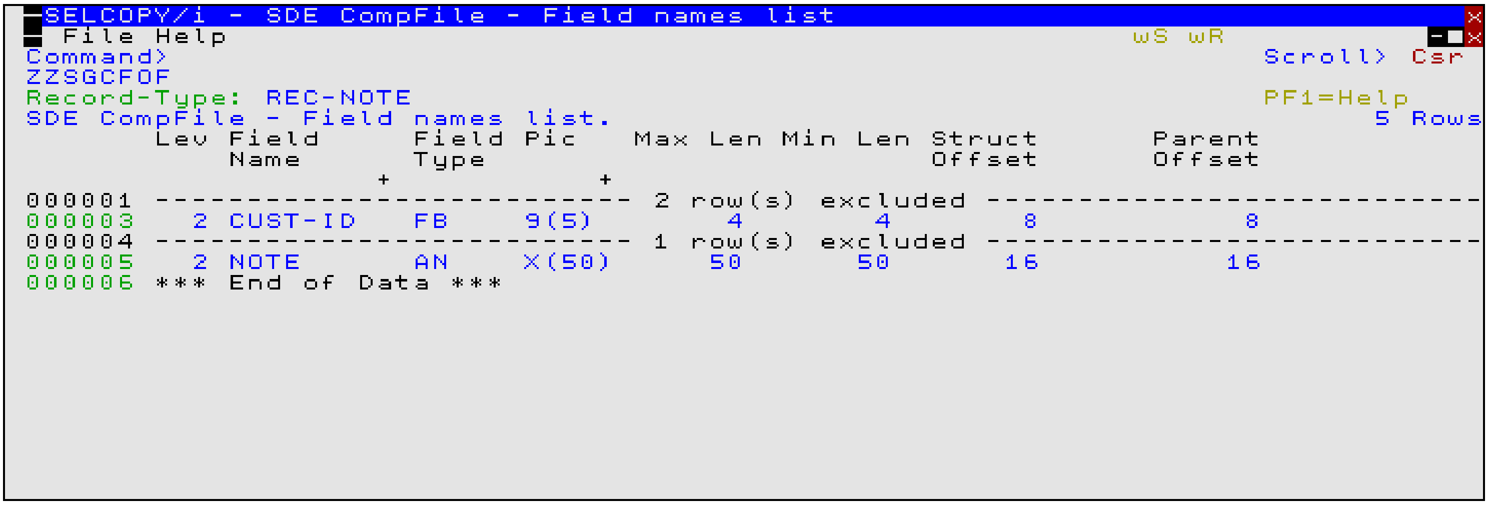
On return from the selectable field list, the **Fields Selected** column will be updated to indicate the number of fields included for compare.

Note that, selecting fields from a record type that has been excluded from the compare in panel Compare Files - Select Record-Types (ZZSGCFOV), will generate the appropriate COMPFILE syntax but will ultimately have no effect. The record type must be included for compare in order for its selected fields to be compared.

Pressing <PF3> to exit the panel, will also save the table of selected record types for which selected fields will be compared, and return to the **Compare Files (formatted): Options** / **Action** panel view.

This panel corresponds to the COMPFILE parameters SELECT *fieldname*, ... FROM *rectype*
Default is to include all field columns of the same name belonging to record types of the same name in *old_structname* and *new_structname*.

SELCOPY/i Compare Files - Select Field Names List



```
 SELCOPY/i - SDE CompFile - Field names list
  File Help                                                   wS wR
Command>                                                           Scroll> Csr
ZZSGCFOF
Record-Type: REC-NOTE                                              PF1=Help
SDE CompFile - Field names list.                                        5 Rows
       Lev Field      Field Pic    Max Len Min Len Struct       Parent
           Name       Type                         Offset       Offset
                   +           +
000001 -------------------------- 2 row(s) excluded ----------------------------
000003   2 CUST-ID    FB    9(5)      4       4       8            8
000004 -------------------------- 1 row(s) excluded ----------------------------
000005   2 NOTE       AN    X(50)    50      50      16           16
000006 *** End of Data ***
```

Figure 53. SELCOPY/i - Compare Files - Select from Field Names List.

The **Compare Files - Select from Field Names List** panel (ZZSGCFOF) is displayed for each record type selected from the Compare Files - Select Field Names to Compare panel.

A list of field names, defined by the selected record type, is presented to the user as an editable table. Standard SELCOPY/i table editing techniques should be used to exclude and re-order the fields so that only fields which are to be compared are displayed in the order in which they are to be compared.

Only included field name entries are compared, therefore a field name may be excluded (as opposed to deleted) in order to exclude it from the compare. This has the benefit that it may easily be included again later if necessary. For example, the following commands may be executed to filter (include) specific table rows:

```
WHERE (length(strip(SelectFld),'T') > 5)  and  (#3 = 'BN')
```

Exclude all rows except those where the length of the Field Name entry is greater than 5 and the Field Picture Type is "BN". Note that the "Field Pic Type" column is field reference number 3.

```
MORE SelectLev < 3
```

Include previously excluded entries where the field level is 1 or 2. (Entries that are already included will remain included.)

The order in which the field names occur is the order in which the fields will be compared. Performance may be improved if the Exclude Changed option is set and fields where differences are expected are specified first in this list.

For Formatted or Hierarchical compare involving sorted or unsorted key synchronisation where key segments are specified as field names, then the key segment field names may be excluded from the compare. This is because the key segment fields must match in order to establish the synchronised record pair prior to comparing the remaining record field data.

Pressing <PF3> to exit the panel, will also save the table of selected field names and return to the **Compare Files - Select Field Names to Compare** panel.

This panel corresponds to the COMPFILE parameters SELECT *fieldname*, ... FROM *rectype*
Default is to include all field columns of the same name belonging to record types of the same name in *old_structname* and *new_structname*.

Compare Files Output

Report Format

The report generated by the compare files utility is a structured data file designed to be browsed (not printed) using a structure definition object (SDO) within a SELCOPY/i session.

The associated SDO is automatically generated when the compare files utility is run. The SDO dataset name is always the DSN of the report with a suffix of "**.SDO**". If the report output is to a PDS/PDSE library member, then the SDO will be written to a library member of the same name belonging to PDS/PDSE library DSN with suffix ".SDO".

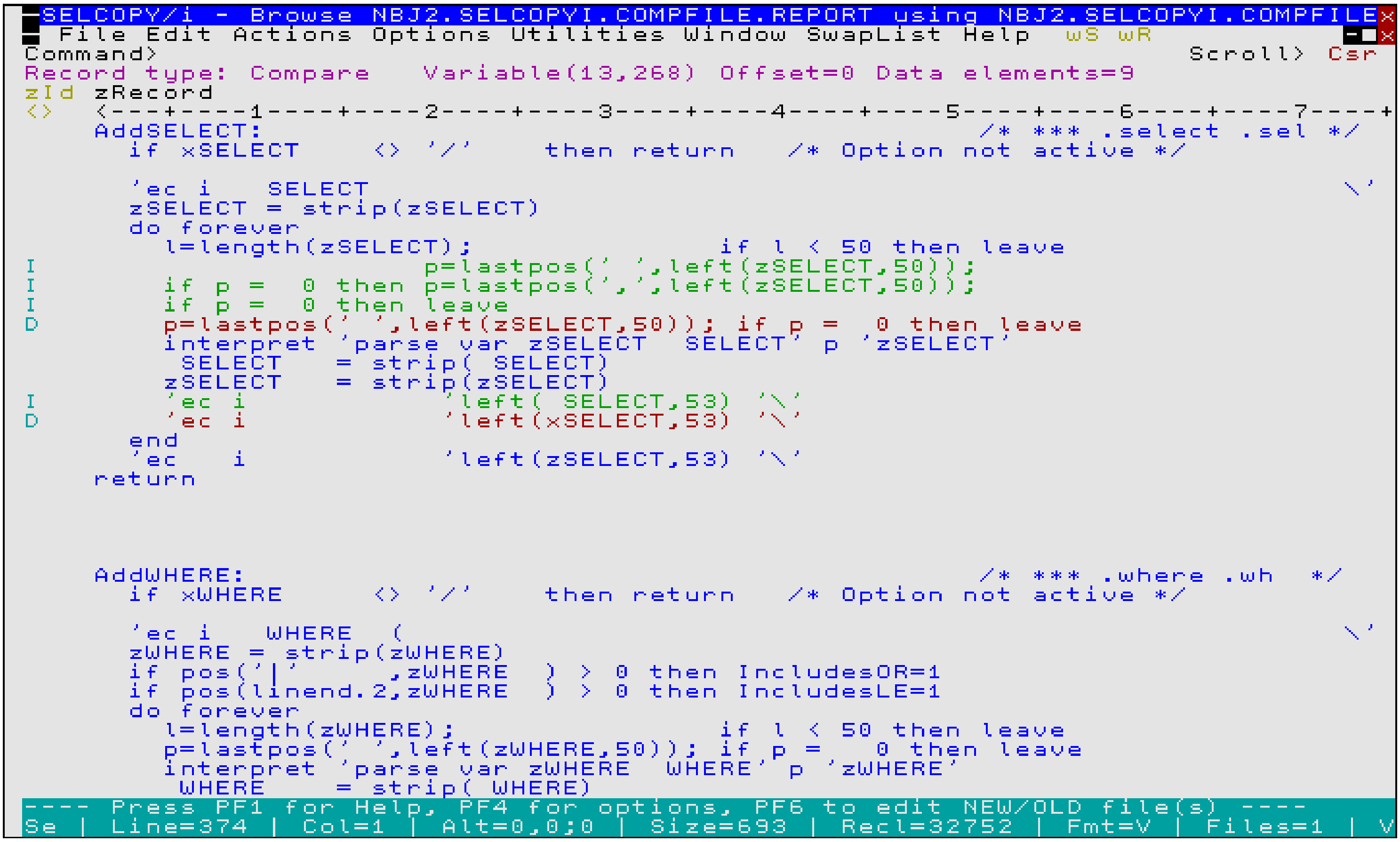
If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

The default DSN for the report and SDO data sets is "*user*.SELCOPYI.COMPFILE.REPORT" and "*user*.SELCOPYI.COMPFILE.REPORT.SDO" respectively.

If the compare process is run in the foreground of a SELCOPY/i session, then the output report file is automatically browsed on completion.

If the compare process is run as a batch job, then the output report file may then be subsequently browsed from your SELCOPY/i session by issuing the command **CFOUT *report_file_name***, or by issuing CFOUT as a prefix command against the report DSN or member name in a dataset list or library list window.

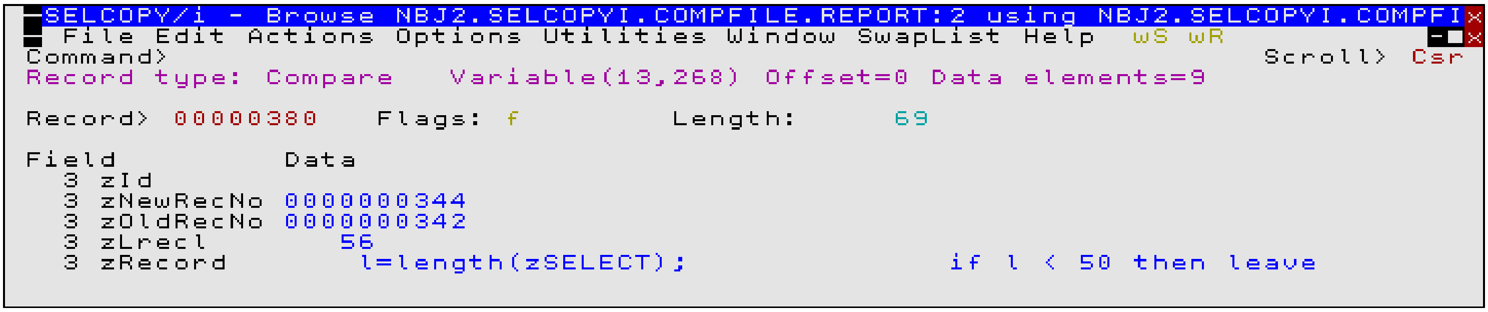
In order to display the more of the record data, when a compare files output report is viewed on an 80-character width 3270 terminal, the prefix area and report record type Compare fields **zNewRecNo**, **zOldRecNo** and **zLrecl** are automatically suppressed from view.



```
 SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT using NBJ2.SELCOPYI.COMPFILE
  File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                            Scroll> Csr
Record type: Compare   Variable(13,268) Offset=0 Data elements=9
zId zRecord
<>   <----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+
    AddSELECT:                                         /* *** .select .sel */
      if xSELECT     <> '/'    then return   /* Option not active */

      'ec i    SELECT                                                        \'
      zSELECT = strip(zSELECT)
      do forever
        l=length(zSELECT);                   if l < 50 then leave
I                      p=lastpos(' ',left(zSELECT,50));
I       if p =  0 then p=lastpos(',',left(zSELECT,50));
I       if p =  0 then leave
D       p=lastpos(' ',left(zSELECT,50)); if p =  0 then leave
        interpret 'parse var zSELECT  SELECT' p 'zSELECT'
         SELECT    = strip( SELECT)
        zSELECT    = strip(zSELECT)
I       'ec i                'left( SELECT,53) '\'
D       'ec i                'left(xSELECT,53) '\'
      end
      'ec   i                'left(zSELECT,53) '\'
    return



    AddWHERE:                                          /* *** .where .wh   */
      if xWHERE      <> '/'    then return   /* Option not active */

      'ec i    WHERE   (                                                     \'
      zWHERE = strip(zWHERE)
      if pos('|'       ,zWHERE   ) > 0 then IncludesOR=1
      if pos(linend.2,zWHERE   ) > 0 then IncludesLE=1
      do forever
        l=length(zWHERE);                    if l < 50 then leave
        p=lastpos(' ',left(zWHERE,50)); if p =   0 then leave
        interpret 'parse var zWHERE  WHERE' p 'zWHERE'
         WHERE     = strip( WHERE)
---- Press PF1 for Help, PF4 for options, PF6 to edit NEW/OLD file(s) ----
Se | Line=374 | Col=1 | Alt=0,0;0 | Size=693 | Recl=32752 | Fmt=V | Files=1 | V
```

Figure 54. SELCOPY/i - Compare Files - Output Report - 80 column screen.

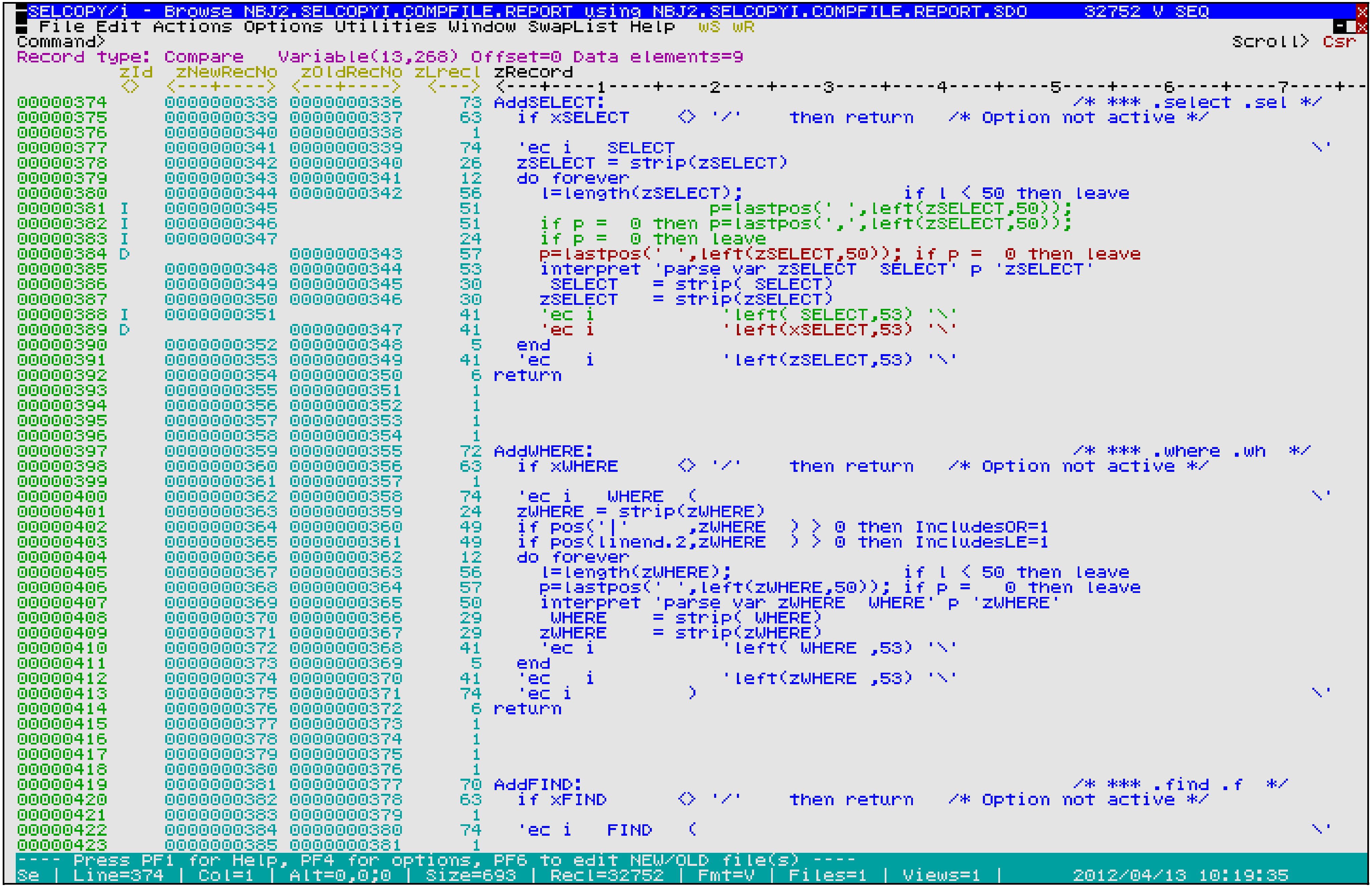
To view these suppressed fields press <PF2> with the cursor on any report record to see it in vertical format, with all fields included. (Further useful function key definitions are detailed below.) Alternativeley, issue the command **SELect *** with the cursor positioned on any record of record type Compare to reveal the suppressed fields. If necessary, the command **PREfix ON n** will provide a prefix area of length *n* (default is 8).



```
 SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT:2 using NBJ2.SELCOPYI.COMPFI
  File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                            Scroll> Csr
Record type: Compare   Variable(13,268) Offset=0 Data elements=9

Record> 00000380    Flags: f          Length:      69

Field           Data
   3 zId
   3 zNewRecNo 0000000344
   3 zOldRecNo 0000000342
   3 zLrecl       56
   3 zRecord       l=length(zSELECT);                 if l < 50 then leave
```

Figure 55. SELCOPY/i - Compare Files - Output Report - Zoomed view.

A compare files output report is viewed on a 3270 terminal of width greater than 80 characters will include a prefix area with no report field suppression.



```
SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT using NBJ2.SELCOPYI.COMPFILE.REPORT.SDO     32752 V SEQ
 File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                                                           Scroll> Csr
Record type: Compare   Variable(13,268) Offset=0 Data elements=9
         zId  zNewRecNo  zOldRecNo zLrecl zRecord
         <>  <---+----> <---+---->  <---> <---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
00000374     0000000338 0000000336     73 AddSELECT:                                         /* *** .select .sel */
00000375     0000000339 0000000337     63   if xSELECT     <> '/'    then return   /* Option not active */
00000376     0000000340 0000000338      1
00000377     0000000341 0000000339     74   'ec i   SELECT                                                        \'
00000378     0000000342 0000000340     26   zSELECT = strip(zSELECT)
00000379     0000000343 0000000341     12   do forever
00000380     0000000344 0000000342     56     l=length(zSELECT);              if l < 50 then leave
00000381 I   0000000345                51                    p=lastpos(' ',left(zSELECT,50));
00000382 I   0000000346                51     if p =  0 then p=lastpos(',',left(zSELECT,50));
00000383 I   0000000347                24     if p =  0 then leave
00000384 D              0000000343     57     p=lastpos(' ',left(zSELECT,50)); if p =  0 then leave
00000385     0000000348 0000000344     53     interpret 'parse var zSELECT  SELECT' p 'zSELECT'
00000386     0000000349 0000000345     30      SELECT   = strip( SELECT)
00000387     0000000350 0000000346     30     zSELECT   = strip(zSELECT)
00000388 I   0000000351                41     'ec i           'left( SELECT,53) '\'
00000389 D              0000000347     41     'ec i           'left(xSELECT,53) '\'
00000390     0000000352 0000000348      5   end
00000391     0000000353 0000000349     41   'ec   i         'left(zSELECT,53) '\'
00000392     0000000354 0000000350      6 return
00000393     0000000355 0000000351      1
00000394     0000000356 0000000352      1
00000395     0000000357 0000000353      1
00000396     0000000358 0000000354      1
00000397     0000000359 0000000355     72 AddWHERE:                                          /* *** .where .wh  */
00000398     0000000360 0000000356     63   if xWHERE      <> '/'    then return   /* Option not active */
00000399     0000000361 0000000357      1
00000400     0000000362 0000000358     74   'ec i   WHERE   (                                                     \'
00000401     0000000363 0000000359     24   zWHERE = strip(zWHERE)
00000402     0000000364 0000000360     49   if pos('|'     ,zWHERE   ) > 0 then IncludesOR=1
00000403     0000000365 0000000361     49   if pos(linend.2,zWHERE   ) > 0 then IncludesLE=1
00000404     0000000366 0000000362     12   do forever
00000405     0000000367 0000000363     56     l=length(zWHERE);               if l < 50 then leave
00000406     0000000368 0000000364     57     p=lastpos(' ',left(zWHERE,50)); if p =   0 then leave
00000407     0000000369 0000000365     50     interpret 'parse var zWHERE  WHERE' p 'zWHERE'
00000408     0000000370 0000000366     29      WHERE    = strip( WHERE)
00000409     0000000371 0000000367     29     zWHERE    = strip(zWHERE)
00000410     0000000372 0000000368     41     'ec i           'left( WHERE ,53) '\'
00000411     0000000373 0000000369      5   end
00000412     0000000374 0000000370     41   'ec   i         'left(zWHERE ,53) '\'
00000413     0000000375 0000000371     74   'ec i           )                                                     \'
00000414     0000000376 0000000372      6 return
00000415     0000000377 0000000373      1
00000416     0000000378 0000000374      1
00000417     0000000379 0000000375      1
00000418     0000000380 0000000376      1
00000419     0000000381 0000000377     70 AddFIND:                                           /* *** .find .f   */
00000420     0000000382 0000000378     63   if xFIND       <> '/'    then return   /* Option not active */
00000421     0000000383 0000000379      1
00000422     0000000384 0000000380     74   'ec i   FIND    (                                                     \'
00000423     0000000385 0000000381      1
---- Press PF1 for Help, PF4 for options, PF6 to edit NEW/OLD file(s) ----
Se | Line=374 | Col=1 | Alt=0,0;0 | Size=693 | Recl=32752 | Fmt=V | Files=1 | Views=1 |      2012/04/13 10:19:35
```

Figure 56. SELCOPY/i - Compare Files - Output Report - wide-screen.

The report file is comprised of several different record types detailed below.

Record Type: Command

`Timestamp`
The date and time (yyyy/mm/dd hh:mm:ss) at which the comparison was run.

`Command`
This is the COMPFILE command syntax generated by the dialog panels that was used to run the compare files utility.

Record Type: Files

`Type`
The file reference type ("New" or "Old").

`Dataset`
The fully qualified dataset name (including library member name) or HFS file path.

Record Type: Compare | Compare-*record_type*

`zID`
This field displays one of the following codes which correspond to the data displayed in the zRecord or *<field1>*, *<field2>*, etc. fields.

(blank)	A record in both the NEW and OLD files that is flagged as being **matched**. Matching record data in these report records are colour coded **BLUE**.
D	An OLD file record that is flagged as having been **deleted** from the NEW file. Deleted records are colour-coded **RED**.
I	A NEW file record that is flagged as having been **inserted**. Inserted records are colour-coded **GREEN**.
CN	A NEW file record flagged as having been changed. Report records with a zID field of "CN" are always followed by a report record with zID field "CO" for the corresponding OLD file record in the record pair. Changed-New records are colour-coded **WHITE**.
CO	An OLD file record flagged as having been changed. Report records with a zID field of "CO" are always preceeded by a report record with zID field "CN" for the coresponding NEW file record in the record pair. Changed-Old records are colour-coded **YELLOW**.
H	Applicable to Basic and Extended Unformatted Compare only, a hilight-changes record follows a CN/CO zID pair of report records, underlining with character "#" (hash) each byte that is different in a record flagged as having been **changed**. Hilight-Changes records are colour-coded **PINK**.

`zNewRecNo`
The NEW file record number. (Field source is 4-byte binary numeric).
By default, this field is suppressed in a table format view of the report on an 80-character width 3270 terminal.

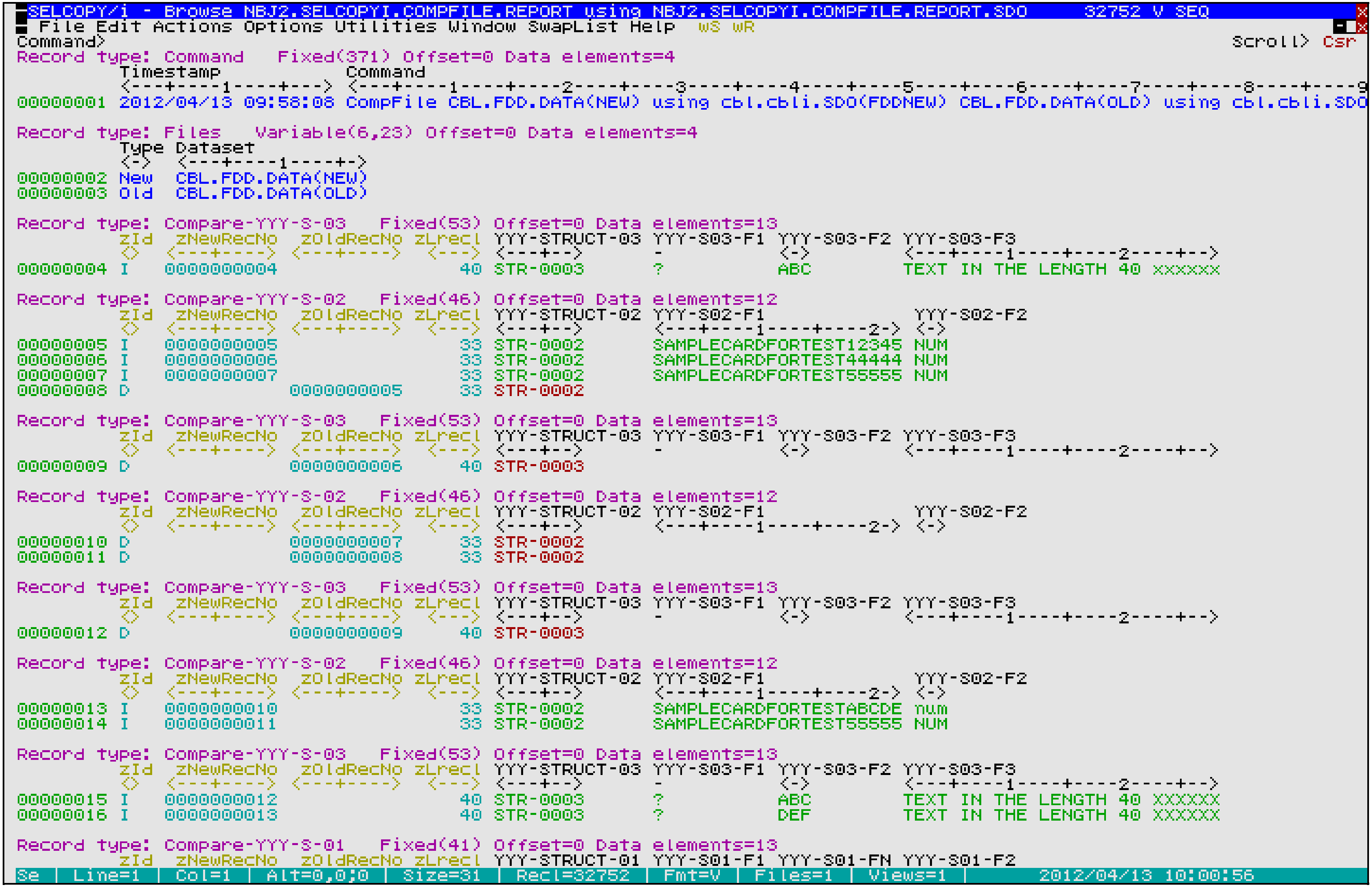
`zOldRecNo`
The OLD file record number. (Field source is 4-byte binary numeric).
By default, this field is suppressed in a table format view of the report on an 80-character width 3270 terminal.

`zLrecl`
The length of the original NEW or OLD file record. (Field source is 2-byte binary numeric).
By default, this field is suppressed in a table format view of the report on an 80-character width 3270 terminal.

`zRecord |` *`<field1> <field2> etc.`*
For Basic and Extended Unformatted compare, the zRecord field is displayed containing the original OLD or NEW file record data.

For Formatted and Hierarchical compare, the zRecord field is replaced with each of the fields selected for compare belonging to the record type assigned to the OLD or NEW file record data.

For formatted NEW file records, the contents of these fields are a faithful copy of the original data. For readability, where NEW and OLD structures are not identical, formatted OLD file records are **remapped** to fit the NEW structure, in order to align like named fields in a table view.



```
 SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT using NBJ2.SELCOPYI.COMPFILE.REPORT.SDO     32752 V SEQ
  File Edit Actions Options Utilities Window SwapList Help   wS wR
Command>                                                                                                   Scroll> Csr
Record type: Command   Fixed(371) Offset=0 Data elements=4
         Timestamp           Command
         <---+----1----+---> <---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9
00000001 2012/04/13 09:58:08 CompFile CBL.FDD.DATA(NEW) using cbl.cbli.SDO(FDDNEW) CBL.FDD.DATA(OLD) using cbl.cbli.SDO

Record type: Files   Variable(6,23) Offset=0 Data elements=4
         Type Dataset
         <->  <---+----1----+->
00000002 New  CBL.FDD.DATA(NEW)
00000003 Old  CBL.FDD.DATA(OLD)

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+----> <---+---->  <---> <---+-->      -          <->        <---+----1----+----2----+-->
00000004 I   0000000004                40 STR-0003      ?          ABC        TEXT IN THE LENGTH 40 xxxxxx

Record type: Compare-YYY-S-02   Fixed(46) Offset=0 Data elements=12
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-02 YYY-S02-F1             YYY-S02-F2
         <>  <---+----> <---+---->  <---> <---+-->      <---+----1----+----2-> <->
00000005 I   0000000005                33 STR-0002      SAMPLECARDFORTEST12345 NUM
00000006 I   0000000006                33 STR-0002      SAMPLECARDFORTEST44444 NUM
00000007 I   0000000007                33 STR-0002      SAMPLECARDFORTEST55555 NUM
00000008 D              0000000005     33 STR-0002

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+----> <---+---->  <---> <---+-->      -          <->        <---+----1----+----2----+-->
00000009 D              0000000006     40 STR-0003

Record type: Compare-YYY-S-02   Fixed(46) Offset=0 Data elements=12
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-02 YYY-S02-F1             YYY-S02-F2
         <>  <---+----> <---+---->  <---> <---+-->      <---+----1----+----2-> <->
00000010 D              0000000007     33 STR-0002
00000011 D              0000000008     33 STR-0002

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+----> <---+---->  <---> <---+-->      -          <->        <---+----1----+----2----+-->
00000012 D              0000000009     40 STR-0003

Record type: Compare-YYY-S-02   Fixed(46) Offset=0 Data elements=12
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-02 YYY-S02-F1             YYY-S02-F2
         <>  <---+----> <---+---->  <---> <---+-->      <---+----1----+----2-> <->
00000013 I   0000000010                33 STR-0002      SAMPLECARDFORTESTABCDE num
00000014 I   0000000011                33 STR-0002      SAMPLECARDFORTEST55555 NUM

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+----> <---+---->  <---> <---+-->      -          <->        <---+----1----+----2----+-->
00000015 I   0000000012                40 STR-0003      ?          ABC        TEXT IN THE LENGTH 40 XXXXXX
00000016 I   0000000013                40 STR-0003      ?          DEF        TEXT IN THE LENGTH 40 XXXXXX

Record type: Compare-YYY-S-01   Fixed(41) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-01 YYY-S01-F1 YYY-S01-FN YYY-S01-F2
Se | Line=1 | Col=1 | Alt=0,0;0 | Size=31 | Recl=32752 | Fmt=V | Files=1 | Views=1 |      2012/04/13 10:00:56
```

Figure 57. SELCOPY/i - Compare Files - Output Report - formatted compare.

Record Type: Field

Report records of this type are only included for Formatted or Hierarchical compare only.

`zID`

This field displays one of the following codes which corresponds to the field data displayed in other report record types.

C	This Field report record identifies a **changed field** in the preceeding Compare-*record_type* report record. A separate Field report record is written for each changed field.
K	This Field report record identifies a **key field segment** in a record type. A separate Field report record is written for each key field segment in each keyed record type. If key segments are specified using absolute position and length (instead of by field name) then Key report record types are displayed instead. Field report records identifying key field segments are displayed following the Summary report records.
S	This Field report record identifies a **field specifically selected for compare** in a record type. If fields in a record type have **not** been specifically selected for compare (COMPFILE syntax SELECT *field* FROM *record_type*), then all fields in the NEW structure record type are selected by default and no "S" Field report record is generated for that record type. A separate Field report record is written for each field specifically selected for caompare in each record type. Field report records identifying compare fields are displayed following the Summary report records.

`zRecType`

Identifies the record type name.

`zFieldName`

Identifies the field name within the record type specified by zRecType.

Record Type: Summary

This report record type provides a report summary of the compare files execution. Only one Summary record exists for any compare files execution.

`SyncType`

This field displays the record pair synchronisation technique used for the compare operation. (**Read-Ahead**, **One-to-One**, **Keyed** or **Keyed (unsorted)**)

`NewRecsTot`

The total number of records processed from the NEW file.

`OldRecsTot`

The total number of records processed from the OLD file.

`Matches`

The total number of NEW and OLD file records that match.

`Changed`

The total number of changed records.

`Deleted`

The total number of OLD file records deleted.

`Inserted`

The total number of NEW file records inserted.

`NewNotSel`

For a formatted compare only, the total number of records from the NEW file that were not selected for comparison because they did not fit any record type selection criteria and so were not assigned a record type.

`OldNotSel`

For a formatted compare only, the total number of records from the OLD file that were not selected for comparison because they did not fit any record type selection criteria and so were not assigned a record type.

Record Type: Key

This report record type displays information about a key specified as absolute record key positions and lengths. A Key report record is displayed for each segment of the key defined this way.

`Length`
The Key segment length.

`NewKeyPos`
The Key segment position in the NEW file.

`OldKeyPos`
The Key segment position in the OLD file.

Function Keys

<PF1>	Display context sensitive help.
<PF2>	Display the report record in a new window in single format (vertical) view. In single format view, use **<PF10>**/**<PF11>** to display the previous/next report record respectively.
<PF4>	Display the **CFUTIL** compare files report, multi-function menu. This includes show and hide of report records based on their type, and show **Changed fields only** in a separate window.
<PF6>	Applicable to report records of record type Compare or Compare-*record_type* only, <PF6> edits the file(s) referenced by the focus report record. The SELCOPY/i text editor is used to edit the file, and the display is scrolled directly to the record number referenced in the report focus record. If the focus is a **Matched** record, then both the OLD and NEW files will be placed in the edit ring, with focus passed to the NEW file.

Compare Libraries (=7.2)

Overview

The Compare Libraries (COMPLIB) utility provides a method of performing an unformatted, 1-to-1 compare of records belonging to selected members of two (NEW and OLD) PDS/PDSE libraries. All record data in selected members of the NEW library are compared with members of the same name in the OLD library.

The compare process utilises the SELCOPY program with input control statements being read from member ZZSCOMPL of the distributed sample library SZZSSAM1. The Compare Libraries utility can execute SELCOPY in the foreground (interactively) or may be used to generate JCL to execute the SELCOPY job in batch.

The SELCOPY report output is written to SYSPRINT so that each report line details the result of the compare between one NEW and OLD library member pair, and also includes a COMPFILE command to compare the two library members individually. When the SYSPRINT output is displayed in a SELCOPY/i text edit view, the generated COMPFILE commands may be executed using the CMDTEXT facility (i.e simply position the cursor on the required COMPFILE command and press the <PF4> key.

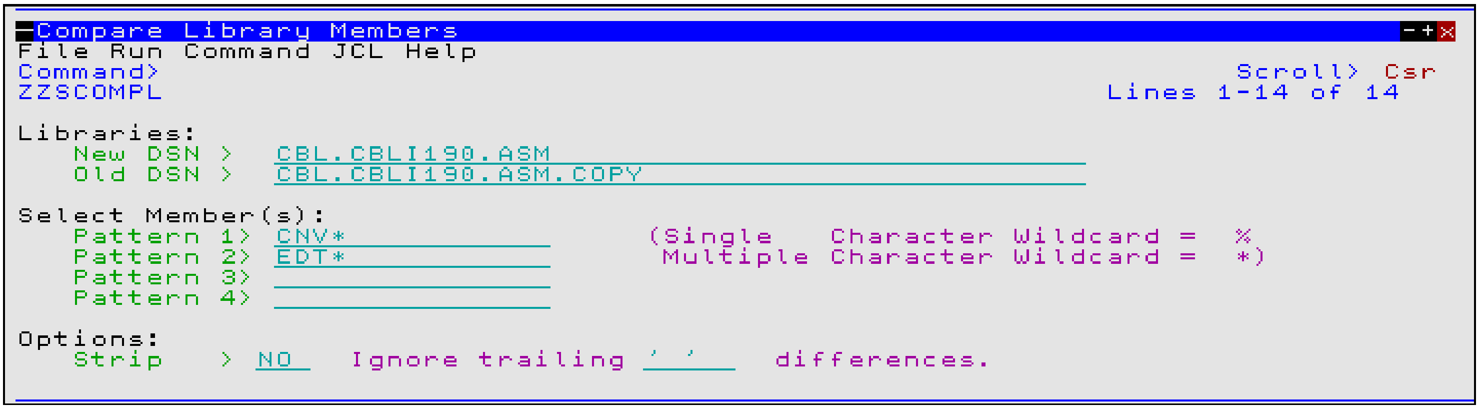
If the Compare Libraries utility is executed in the foreground, then the result of each member compare is logged to the terminal and the SYSPRINT output presented to the user in a temporary text edit view.

Compare Libraries Panel

The Compare Library Members utility panel window (ZZSCOMPL) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select 'Compare Libraries' from the Utilities menu.
- Execute the command COMPLIB with no parameters from the command line of any window.
- Execute the prefix command "**CL**" against a PDS/PDSE library DSN entry of a file List type window. The resulting Compare Library Members panel window will treat the corresponding list entry as the New DSN.

By default, field entries are populated with arguments and options that were entered the last time the Compare Library Members Utility panel was used.



```
 Compare Library Members                                                  -+x
File Run Command JCL Help
Command>                                                           Scroll> Csr
ZZSCOMPL                                                    Lines 1-14 of 14

Libraries:
   New DSN >   CBL.CBLI190.ASM
   Old DSN >   CBL.CBLI190.ASM.COPY

Select Member(s):
   Pattern 1> CNV*                 (Single   Character Wildcard =  %
   Pattern 2> EDT*                  Multiple Character Wildcard =  *)
   Pattern 3>
   Pattern 4>

Options:
   Strip    > NO    Ignore trailing ' '     differences.
```

Figure 58. Compare Library Members Panel.

Having typed entries in the required panel fields, simply pressing the <Enter> key or, if configured, double-clicking the left mouse button will will action the library compare in the foreground.

Alternatively, the user may select an item from the menu bar.

Menu Bar Items

Run

Run the library compare in the foreground.

Command

Generate the COMPLIB command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using CMDTEXT point-and-shoot execution <PF4> or copied into the user's HOME file and saved for future execution.

JCL

Generate a JCL job stream that executes the **SELCOPY** program. The SYSIN input comprises the SELCOPY control statements member ZZSCOMPL and a list of member names that match the specified member patterns. The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Panel Input Fields

New DSN>

This input field (LIBRARY1) is mandatory and identifies the NEW PDS or PDSE library DSN to be compared.

Old DSN>

This input field (LIBRARY2) is mandatory and identifies the OLD PDS or PDSE library DSN to be compared.

Select Members(s):
Pattern 1>
Pattern 2>
Pattern 3>
Pattern 4>

These input fields (MEMBER1, MEMBER2, MEMBER3 and MEMBER4) are optional and allow the user to provide up to 4 alternative member name masks for selecting members to be compared from the **NEW library**. If a NEW library member name matches any of these masks, then it will be selected for compare.

Member names in the OLD library that match this mask but do not exist in the NEW library will **not** be included for compare.

A member name mask supports the following wild cards:

*	A single asterisk represents an entire member name or zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If no member name masks are specified, then all members of the NEW **and** OLD libraries will be compared.

`Strip>`

This input field (STRIP) contains either "YES" or "NO" and indicates whether trailing characters that match the specified strip character, are to be stripped from the longer record, to the length of the shorter record, when the records to be compared are of different lengths.

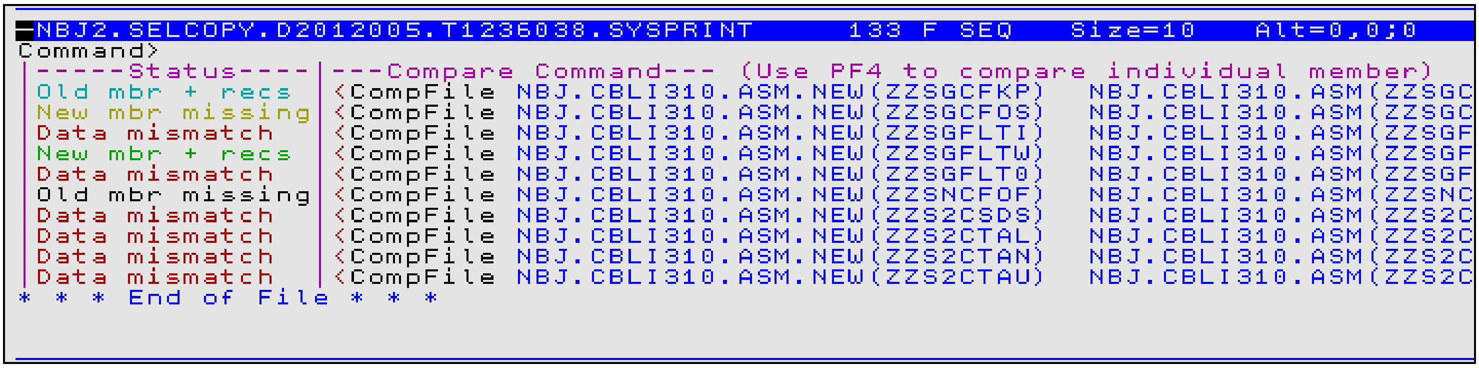
`Ignore trailing '`*`char`*`' differences.`

This input field (STRIPC) specifies the strip character to be used if the STRIP field conmtains "YES".

Compare Libraries Output

The output generated by the Compare Libraries utility is a report of the compare on each NEW and OLD library member pair for which the data did not match. This report is displayable EBCDIC text and may be viewed using the SELCOPY/i (CBLe) text editor.

Each line of the output report corresponds to a single member compare.



```
 NBJ2.SELCOPY.D2012005.T1236038.SYSPRINT        133 F SEQ    Size=10    Alt=0,0;0
Command>
 |-----Status----|---Compare Command--- (Use PF4 to compare individual member)
 |Old mbr + recs |<CompFile NBJ.CBLI310.ASM.NEW(ZZSGCFKP)  NBJ.CBLI310.ASM(ZZSGC
 |New mbr missing|<CompFile NBJ.CBLI310.ASM.NEW(ZZSGCFOS)  NBJ.CBLI310.ASM(ZZSGC
 |Data mismatch  |<CompFile NBJ.CBLI310.ASM.NEW(ZZSGFLTI)  NBJ.CBLI310.ASM(ZZSGF
 |New mbr + recs |<CompFile NBJ.CBLI310.ASM.NEW(ZZSGFLTW)  NBJ.CBLI310.ASM(ZZSGF
 |Data mismatch  |<CompFile NBJ.CBLI310.ASM.NEW(ZZSGFLT0)  NBJ.CBLI310.ASM(ZZSGF
 |Old mbr missing|<CompFile NBJ.CBLI310.ASM.NEW(ZZSNCFOF)  NBJ.CBLI310.ASM(ZZSNC
 |Data mismatch  |<CompFile NBJ.CBLI310.ASM.NEW(ZZS2CSDS)  NBJ.CBLI310.ASM(ZZS2C
 |Data mismatch  |<CompFile NBJ.CBLI310.ASM.NEW(ZZS2CTAL)  NBJ.CBLI310.ASM(ZZS2C
 |Data mismatch  |<CompFile NBJ.CBLI310.ASM.NEW(ZZS2CTAN)  NBJ.CBLI310.ASM(ZZS2C
 |Data mismatch  |<CompFile NBJ.CBLI310.ASM.NEW(ZZS2CTAU)  NBJ.CBLI310.ASM(ZZS2C
* * * End of File * * *
```

Figure 59. Compare Libraries Output.

Each output report record is comprised of the following fields:

Status

The status identifies the cause of the unsuccessful member data compare.

Data mismatch

A difference was identified in at least one record of the library members.

On encountering a difference in the record data, no further record matching occurs for that member. Note that the members may also contain a different number of records but reporting the data mismatch takes precedence.

Old mbr + recs

Data in the NEW and OLD library member pair matches, however, additional records were found in the OLD library member.

New mbr + recs

Data in the NEW and OLD library member pair matches, however, additional records were found in the NEW library member.

Old mbr missing

No member exists in the OLD library that matches the NEW library member name.

New mbr missing

No member exists in the NEW library that matches the OLD library member name.

Note that, if a member mask has been used to select a subset of members from the NEW library, then this report status will never occur.

Compare Command

The Compare Command field contains a COMPFILE command which may be executed to generate a more detailed report of the differences that exist between the pair of library members. COMPFILE involes the Compare Files utility.

This command syntax also identifies the NEW and OLD library members of the same member name to which the report line refers.

To execute the COMPFILE command directly from the edited report, simply position the cursor on the command and press the <PF4> key.

SELCOPY Debug & Development (=8.1)

SELCOPY/i can invoke the SELCOPY program allowing interactive execution and debug within SELCOPY/i windows.

- Invoking SELCOPY Debug
- Load Library Search Chain
- SELCOPY Debug Windows
- SELCOPY Debug Commands
- SELCOPY Debug Function Keys

Invoking SELCOPY Debug

The SELCOPY Debug application window is a special instance of the CBLe Text Edit application and always opens as an MDI frame window within the SELCOPY/i main window display area (i.e not as a child window of the invoking CBLe Text Edit main window.) See SELCOPY Debug Main Window for information on the SELCOPY Debug window environment.

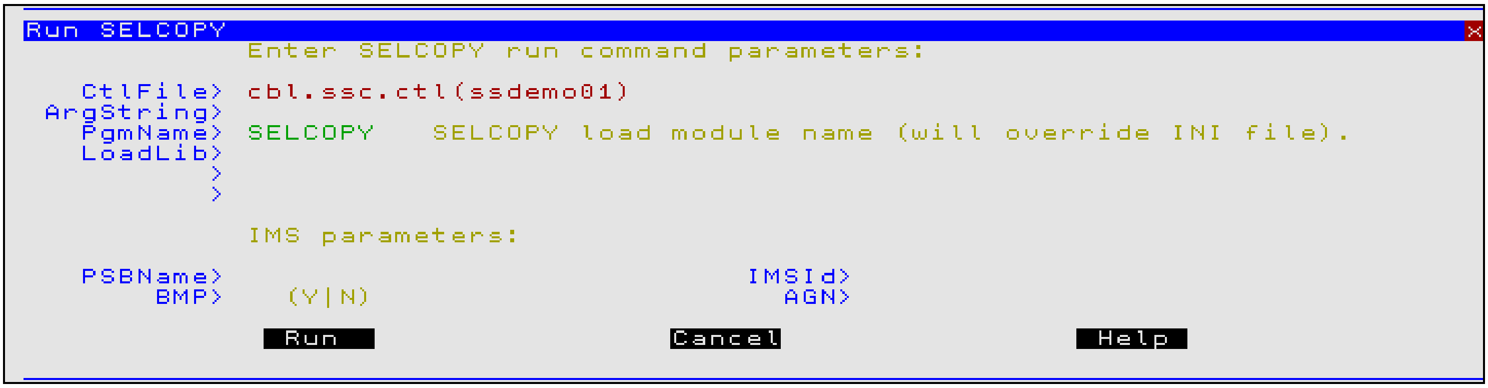
SELCOPY Debug is started via the following:

- Select option 1. 'SELCOPY/debug' from the Utilities Menu
- Select 'SELCOPY Debug/Dev' from the File menu in the CBLe main window menu bar.
- Enter the command SELCOPY on the command line of any window.

See also the CBLe macro, JCLCMX , which is a pre-processor tool used to convert MVS JCL decks containing SELCOPY steps into a format suitable for input to SELCOPY Debug. Simply edit the JCL deck using CBLe, then key in JCLCMX from the edit command line to generate command files containing ALLOC commands that correspond to the DD statements; CALL commands that correspond to EXEC statements; and SELCOPY commands that correspond to EXEC PGM=SELCOPY statements.

A control file name containing the SELCOPY source statements must be provided as SYSIN/SYSIPT input to the SELCOPY Debug window either as a parameter on the SELCOPY command or via the "Run SELCOPY" dialog window. The control file may be specified as a complete fileid (DSN) or as a previously allocated filename (DD/FILEDEF/DLBL) which may be a concatenation of data sets. **e.g.**

```
selcopy  -ctl cbl.ssc.ctl(ssdemo01)
```



```
Run SELCOPY
             Enter SELCOPY run command parameters:

   CtlFile>  cbl.ssc.ctl(ssdemo01)
 ArgString>
   PgmName>  SELCOPY   SELCOPY load module name (will override INI file).
   LoadLib>
          >
          >

             IMS parameters:

   PSBName>                                IMSId>
       BMP>    (Y|N)                         AGN>
```

Figure 60. Run SELCOPY Dialog Window.

If the specified control file is empty or does not yet exist, then SELCOPY Debug opens an empty SYSIN/SYSIPT CBLe text edit view and, on performing its initial control statement analysis, reports ERROR 14 "NO INPUT FILE" in the SYSPRINT/SYSLST view with pop-up message:

```
SDB002E SELCOPY has ended with control card errors. Return Code 52.
```

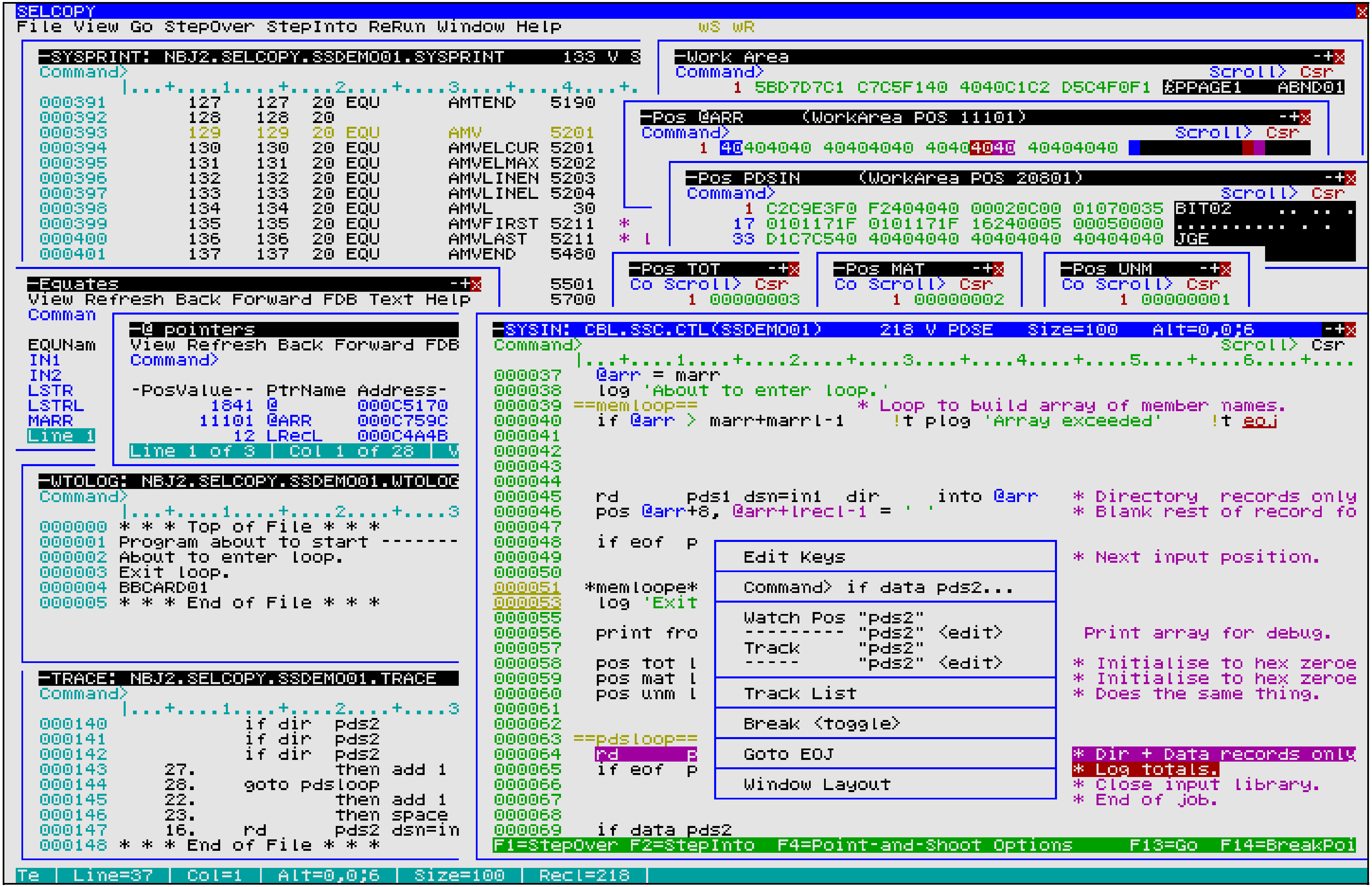
Having selected OK to continue, the user may proceed by adding SELCOPY control statement records to the control file SYSIN/SYSIPT window. When complete, the changes to the control file should be saved before executing RERUN to begin debugging the new SELCOPY statements. If the SELCOPY control fileid does not already exist, then for MVS systems, the Allocate NonVSAM dialog will be opened before the save is actioned.

When SELCOPY Debug starts, the Execute SELCOPY main window is opened together with a CBLe edit view for the control statement SYSIN/SYSIPT input. If the SYSIN/SYSIPT file name was specified as an explicit fileid (DSN) which is not locked (ENQ'd) by another process and the user has sufficient authority, the file is edited Read/Write. Where the SYSIN/SYSIPT input file name was specified as a previously allocated filename (DD/FILEDEF/DLBL), then the data is edited Read/Only.

The SELCOPY program is then executed and stopped following control statement analysis but before actioning the first executable statement. The user may then begin debugging and stepping through the statements.

Note that interactive execution of SELCOPY using statement stepping and/or break points is not supported if the SELCOPY control file contains the SELCOPY option **NOPRINT**, **NOP** or **NOPCTL** to suppress print of the SELCOPY control statements. If any of these options are specified prior to the first control statement, then the job will run to completion without stopping.

Any required SELCOPY input or output files must be allocated before execution of SELCOPY Debug. This excludes SYSIN and SYSPRINT which are handled by the SELCOPY/i window management software. See the CBLe ALLOCATE command and, where the SELCOPY statements form part of an MVS batch stream, use the CBLe JCLCMX REXX macro to configure the environment prior to starting SELCOPY Debug.



```
SELCOPY
File View Go StepOver StepInto ReRun Window Help                 wS wR

-SYSPRINT: NBJ2.SELCOPY.SSDEMO01.SYSPRINT          133 V S
Command>
        |...+....1....+....2....+....3....+....4....+.
000391        127    127   20 EQU        AMTEND     5190
000392        128    128   20
000393        129    129   20 EQU        AMV        5201
000394        130    130   20 EQU        AMVELCUR  5201
000395        131    131   20 EQU        AMVELMAX  5202
000396        132    132   20 EQU        AMVLINEN  5203
000397        133    133   20 EQU        AMVLINEL  5204
000398        134    134   20 EQU        AMVL         30
000399        135    135   20 EQU        AMVFIRST  5211   *
000400        136    136   20 EQU        AMVLAST   5211   * l
000401        137    137   20 EQU        AMVEND    5480
                                                    5501
                                                    5700

-Work Area                                                    -+x
Command>                                               Scroll> Csr
      1 5BD7D7C1 C7C5F140 4040C1C2 D5C4F0F1 $PPAGE1  ABND01

-Pos @ARR          (WorkArea POS 11101)                -+x
Command>                                          Scroll> Csr
      1 40404040 40404040 40404040 40404040

-Pos PDSIN         (WorkArea POS 20801)                   -+x
Command>                                              Scroll> Csr
         1 C2C9E3F0 F2404040 00020C00 01070035 BIT02    .. .. .
        17 0101171F 0101171F 16240005 00050000 .......... . .
        33 D1C7C540 40404040 40404040 40404040 JGE

-Pos TOT      -+x    -Pos MAT      -+x    -Pos UNM      -+x
Co Scroll> Csr       Co Scroll> Csr       Co Scroll> Csr
       1 00000003           1 00000002           1 00000001

-Equates                                     -+x
View Refresh Back Forward FDB Text Help
Comman

EQUNam
IN1
IN2
LSTR
LSTRL
MARR
Line 1

-@ pointers
View Refresh Back Forward FDB
Command>

-PosValue-- PtrName Address-
      1841 @        000C5170
     11101 @ARR     000C759C
        12 LRecL    000C4A4B
Line 1 of 3 | Col 1 of 28 | V

-WTOLOG: NBJ2.SELCOPY.SSDEMO01.WTOLOG
Command>
        |...+....1....+....2....+....3
000000 * * * Top of File * * *
000001 Program about to start -------
000002 About to enter loop.
000003 Exit loop.
000004 BBCARD01
000005 * * * End of File * * *

-TRACE: NBJ2.SELCOPY.SSDEMO01.TRACE
Command>
        |...+....1....+....2....+....3
000140                if dir   pds2
000141                if dir   pds2
000142                if dir   pds2
000143        27.              then add 1
000144        28.     goto pdsloop
000145        22.              then add 1
000146        23.              then space
000147        16.     rd       pds2 dsn=in
000148 * * * End of File * * *

-SYSIN: CBL.SSC.CTL(SSDEMO01)          218 V PDSE    Size=100    Alt=0,0;6    -+x
Command>                                                                Scroll> Csr
       |...+....1....+....2....+....3....+....4....+....5....+....6....+....
000037   @arr = marr
000038   log 'About to enter loop.'
000039 ==memloop==                    * Loop to build array of member names.
000040   if @arr > marr+marrl-1      !t plog 'Array exceeded'      !t eoj
000041
000042
000043
000044
000045   rd        pds1 dsn=in1  dir      into @arr     * Directory records only
000046   pos @arr+8, @arr+lrecl-1 = ' '                 * Blank rest of record fo
000047
000048   if eof  p                                      * Next input position.
000049
000050
000051  *memloope*
000053   log 'Exit
000055
000056   print fro                                       Print array for debug.
000057
000058   pos tot l                                      * Initialise to hex zeroe
000059   pos mat l                                      * Initialise to hex zeroe
000060   pos unm l                                      * Does the same thing.
000061
000062
000063 ==pdsloop==
000064   rd      p                                      * Dir + Data records only
000065   if eof  p                                      * Log totals.
000066                                                  * Close input library.
000067                                                  * End of job.
000068
000069   if data pds2
F1=StepOver F2=StepInto  F4=Point-and-Shoot Options     F13=Go  F14=BreakPoi

   Edit Keys
   Command> if data pds2...
   Watch Pos  "pds2"
   ---------- "pds2"  <edit>
   Track      "pds2"
   ------     "pds2"  <edit>
   Track List
   Break <toggle>
   Goto EOJ
   Window Layout

Te | Line=37 | Col=1 | Alt=0,0;6 | Size=100 | Recl=218 |
```

Figure 61. SELCOPY Debug in 58x120 3270 Session.

Under MVS, interactive execution of IMS SELCOPY programs in DLI and BMP region types is supported.
The following SELCOPY Debug attributes relating to IMS SELCOPY programs may be specified as a parameter on the SELCOPY command or via the "Run SELCOPY" dialog window.

- The PSB name.
- The IMS region (either DLI or BMP). Default is DLI.
- The Subsystem name of the IMS Region to which the connection should be made. **(Optional)**
- IMS Application Group name. **(Optional)**

For IMS DLI regions, IEFRDER, DFSVSAMP must either be allocated or included in the SELCOPY/i System or User INI file for dynamic allocation by SELCOPY Debug at execution time.

Load Library Search Chain

The location of the SELCOPY program executed and any routines called by the SELCOPY operation, CALL, is determined by the standard search chain for the current environment.

Note: The SELCOPY CALL operation is used to pass control to an external Assembler, COBOL routine or any MVS program module developed using Language Environment.

For MVS systems only, SELCOPY Debug provides users with the ability to include additional libraries to the start of the search chain. This gives the SELCOPY Debug environment an equivalent to the STEPLIB JCL statement, which may occur in SELCOPY batch jobs.

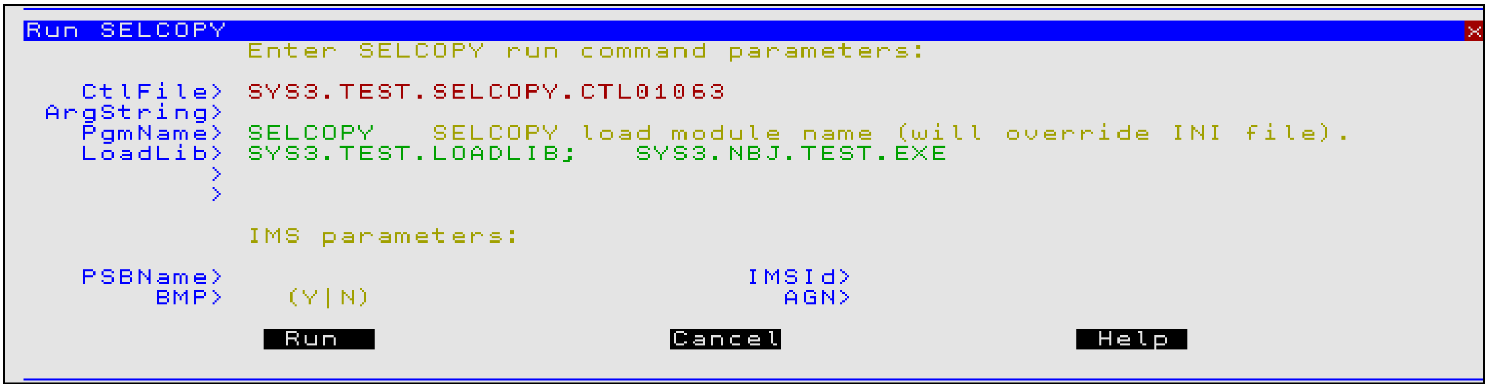
The included library path may be entered in the "Run SELCOPY" dialog window or via the -LIB parameter on the SELCOPY line command, as one of the following:

- A DDname which has been pre-allocated to one or more load libraries.
- One or more load library DSNs separated by ',' (commas), ';' (semi-colons) or ' ' (blanks).

Note: If the DSNs are separated by blanks, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

The following SELCOPY line command and "Run SELCOPY" dialog examples illustrate use of LibPath.

```
  selcopy  -ctl INCTL  -lib "SYS3.TEST.LOADLIB    SYS3.NBJ.TEST.EXE"
```



```
Run SELCOPY
                 Enter SELCOPY run command parameters:

    CtlFile> SYS3.TEST.SELCOPY.CTL01063
  ArgString>
    PgmName> SELCOPY    SELCOPY load module name (will override INI file).
    LoadLib> SYS3.TEST.LOADLIB;    SYS3.NBJ.TEST.EXE
           >
           >

                 IMS parameters:

    PSBName>                                  IMSId>
        BMP>   (Y|N)                            AGN>
```

Figure 62. Run SELCOPY Dialog Window with Load Library Path.

This feature is exploited by the JCLCMX CBLe REXX macro which translates JCL statements from a batch job into the equivalent SELCOPY/i commands, thus setting up an environment suitable for interactive execution of a SELCOPY batch job.

Where STEPLIB JCL statements are found in EXEC PGM=SELCOPY steps, the library names are included on a -LIB parameter in the generated, equivalent SELCOPY command.

SELCOPY Loop Break-in

The nature of SELCOPY execution is such that statements are executed sequentially, or as directed by logic flow operations (e.g. GOTO, PERFORM), until either the last control statement of the SYSIN/SYSIPT input is encountered or a GOTO GET operation is executed.

When one of these conditions occur and at least one input (e.g. READ) and one output (e.g. WRITE, PRINT, UPDATE) operation exists, then processing is passed back to the first run-time control statement in the control deck.

This looping through the control statements will continue until one of the following occurs:

1. End-of-File condition is encountered following an attempted READ of the **prime** input file and no IF EOF condition exists for the file.
2. No further output operations are eligable for execution as a result of a explicit or implicit STOPAFT value. e.g. STOPAFT=50 is implied for LOG output operations and STOPAFT=1 is implied for operations executed based on a true IF INCOUNT condition for equality.
3. GOTO EOJ or GOTO CANCEL operation is executed.
4. A Selection Time Error is encountered.

If none of these conditions occur, then it is possible to introduce an infinite loop in SELCOPY control statement processing.

SELCOPY Debug may be used to identify the cause of this situation or any sequence of statements that cause the control statement stream to loop. It is possible, however, that the user may not know that the loop condition exists until SELCOPY processing has been restarted without a break point in which case, since SELCOPY is executing in the foreground, the 3270 session becomes unresponsive.

It is for this reason that the SELCOPY default break-in facility exists to allow the user to pre-define a default number of times that any control statement within the SELCOPY job stream may be executed before a virtual break point is encountered and processing is paused.

This break-in threshold is controlled via the SELCOPY/i INI option **SELCOPY.LoopBreakIn** which has a default value of 1,000,000.

When the break-in threshold has been reached, a pop-up message window is opened and control is passed back to the user to continue debug investigation. This means that there is no need to forcibly end the SELCOPY/i session and restart the SELCOPY debug process.

Note that a loop break-in may occur even though a loop is not infinite. (e.g. the prime input file may have a number of records greater than the break-in threshold.)

SELCOPY Debug Windows

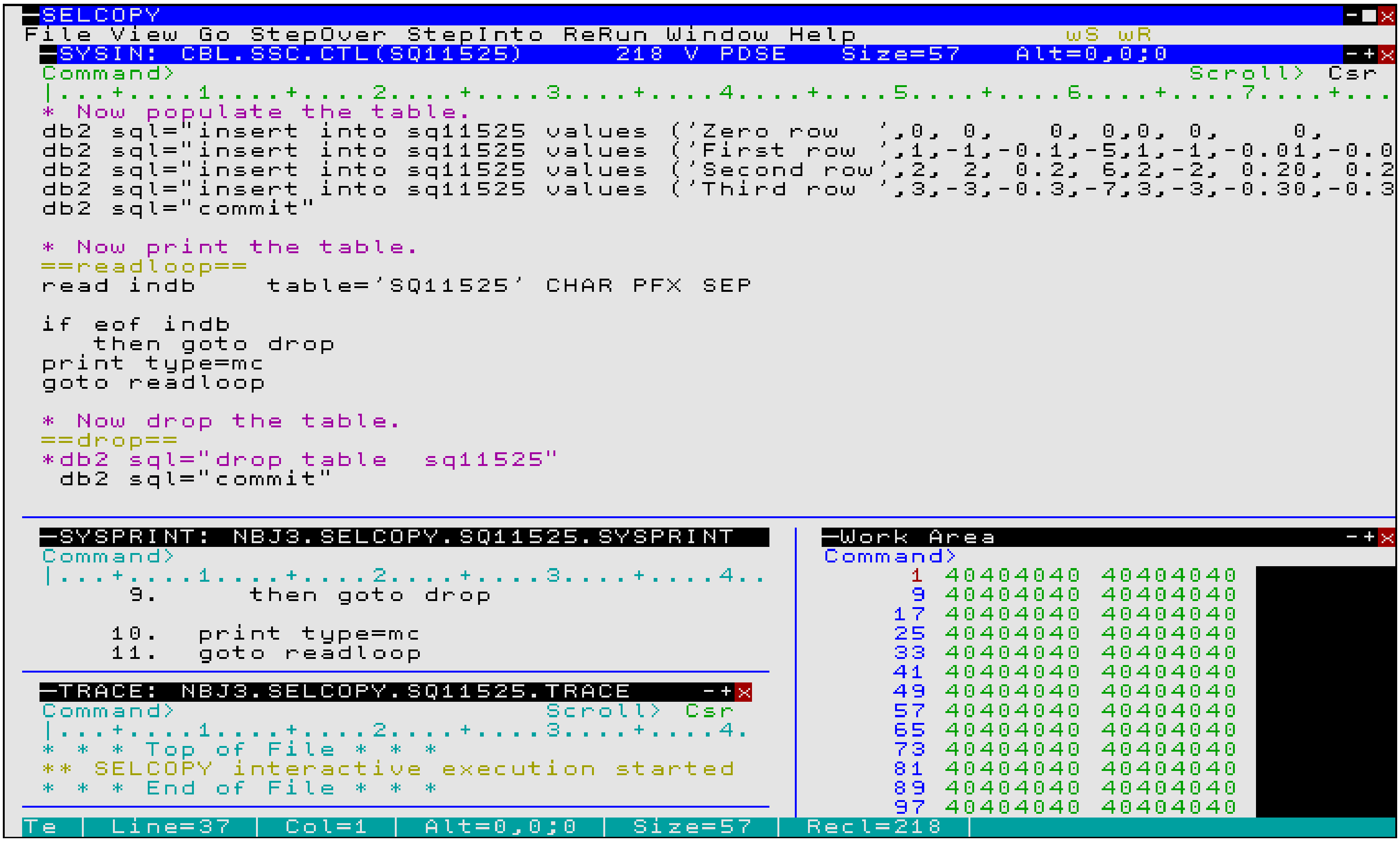
SELCOPY Debug Main window

Like the CBLe text editor, SELCOPY Debug is an MDI (Multiple Document Interface) application. An MDI application comprises a parent (frame) window with a menu bar and a client area within which one or more MDI child windows are displayed. All MDI child windows are confined to the parent window's client area.

The SELCOPY Debug Main (frame) Window supports all MDI child windows supported by the CBLe frame window (including SDE Edit). The SELCOPY Debug frame window is actually a CBLe frame window with additional features and characteristics specifically relating to SELCOPY execution. These features are discussed in this section whereas details on CBLe frame window features may be found in the CBLe Text Edit documentation.

The SELCOPY Debug Main window must always contain the Control Cards, Output Listing and TRACE Windows. Closing any of these windows will quit the SELCOPY main window and so end the Debug session.

When a session is started, these 3 child windows are automatically opened, together with a work area storage window, at fixed locations within the main window client area. The position and size of each window have been pre-determined so that the contents of each window are easily visible when used with terminals of width greater than 80 bytes. Where the terminal display is of width less than 80 bytes, the SELCOPY Debug child windows are opened in a maximised state, however, these may be subsequently restored, resized and repostioned as in Figure 62.



```
SELCOPY
File View Go StepOver StepInto ReRun Window Help                    wS wR
 SYSIN: CBL.SSC.CTL(SQ11525)      218 V PDSE    Size=57    Alt=0,0;0
Command>                                                          Scroll> Csr
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
* Now populate the table.
db2 sql="insert into sq11525 values ('Zero row  ',0, 0,   0, 0,0, 0,    0,
db2 sql="insert into sq11525 values ('First row ',1,-1,-0.1,-5,1,-1,-0.01,-0.0
db2 sql="insert into sq11525 values ('Second row',2, 2, 0.2, 6,2,-2, 0.20, 0.2
db2 sql="insert into sq11525 values ('Third row ',3,-3,-0.3,-7,3,-3,-0.30,-0.3
db2 sql="commit"

* Now print the table.
==readloop==
read indb    table='SQ11525' CHAR PFX SEP

if eof indb
   then goto drop
print type=mc
goto readloop

* Now drop the table.
==drop==
*db2 sql="drop table  sq11525"
 db2 sql="commit"

SYSPRINT: NBJ3.SELCOPY.SQ11525.SYSPRINT      Work Area
Command>                                     Command>
|...+....1....+....2....+....3....+....4..          1 40404040 40404040
      9.     then goto drop                         9 40404040 40404040
                                                   17 40404040 40404040
     10.   print type=mc                           25 40404040 40404040
     11.   goto readloop                           33 40404040 40404040
                                                   41 40404040 40404040
TRACE: NBJ3.SELCOPY.SQ11525.TRACE                  49 40404040 40404040
Command>                            Scroll> Csr    57 40404040 40404040
|...+....1....+....2....+....3....+....4.          65 40404040 40404040
* * * Top of File * * *                            73 40404040 40404040
** SELCOPY interactive execution started           81 40404040 40404040
* * * End of File * * *                            89 40404040 40404040
                                                   97 40404040 40404040
Te | Line=37 | Col=1 | Alt=0,0;0 | Size=57 | Recl=218 |
```

Figure 63. SELCOPY Main Window in 43x80 3270 Session - Resized Child Windows.

Note that the "Ws" (Window Save) button may be used to save a focus child window's size and location within the parent window so that it may subsequently be restored using the "Wr" (Window Restore) button. This enables the user to maintain preferred window size and location across invocations of SELCOPY Debug.

All SELCOPY Debug child windows, other than list and storage windows, are CBLe text edit windows (i.e. Control Cards, Output Listing, trace and log windows.) This allows the user to edit the data in these windows and to issue CBLe commands and macros such as ALL, LOCATE, CHANGE and SAVE.

In addition to the standard SELCOPY Debug windows, the user can open a CBLe edit view for any other file (e.g. the input data sets, etc.), thus giving SELCOPY Debug all the features provided by the CBLe text editor. Also, if MDILIST is ON (default), any LIST window opened from a SELCOPY Debug child window will itself be a child window of SELCOPY Debug.

By default, PF9 is assigned to line command, MDINext, and is used to pass focus between the SELCOPY Debug child windows.

SYSIN Window

The SYSIN window is opened automatically when SELCOPY Debug is started. It may also be opened via the following:

- Select 'Control Cards' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW CTL.

The SYSIN window is an edit view that contains the control statement source file as required for execution of SELCOPY Debug. This window highlights the current operation and allows the user to set and unset break points.

By default, SELCOPY Debug attempts to edit the SYSIN file read/write. If this is not possible, the user is prompted to continue the session with the file edited in read only mode. In either case, the edit profile macro is executed when the file is loaded. If the CBL supplied macro PROFILE is set as the default edit profile, then useful edit buttons are added to the menu bar. See the PROFILE and PROFIRST macros for a description of each button's use.

Note that SELCOPY analyses the control statements prior to execution and it is at this point that SELCOPY Debug associates each operation in the SYSIN display with its appropriate selection id. Therefore, any alterations made to the SYSIN data during SELCOPY debugging must first be saved and the job re-started before any further statement execution can take place.

The contents of the window scroll automatically in order to display the current statement in the SELCOPY execution. As for any edit view, CBLe commands and macros may be used to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)

In addition to any CBLe edit highlighting, during the course of execution control statements are highlighted as follow:

1. Next executable SELCOPY statement. Default highlight - pink reverse video.
2. Break Point. Default highlight - red reverse video.

Closing the SYSIN window also exits SELCOPY Debug.

SYSPRINT Window

The SYSPRINT window is opened automatically when SELCOPY Debug is started. It may also be opened via the following:

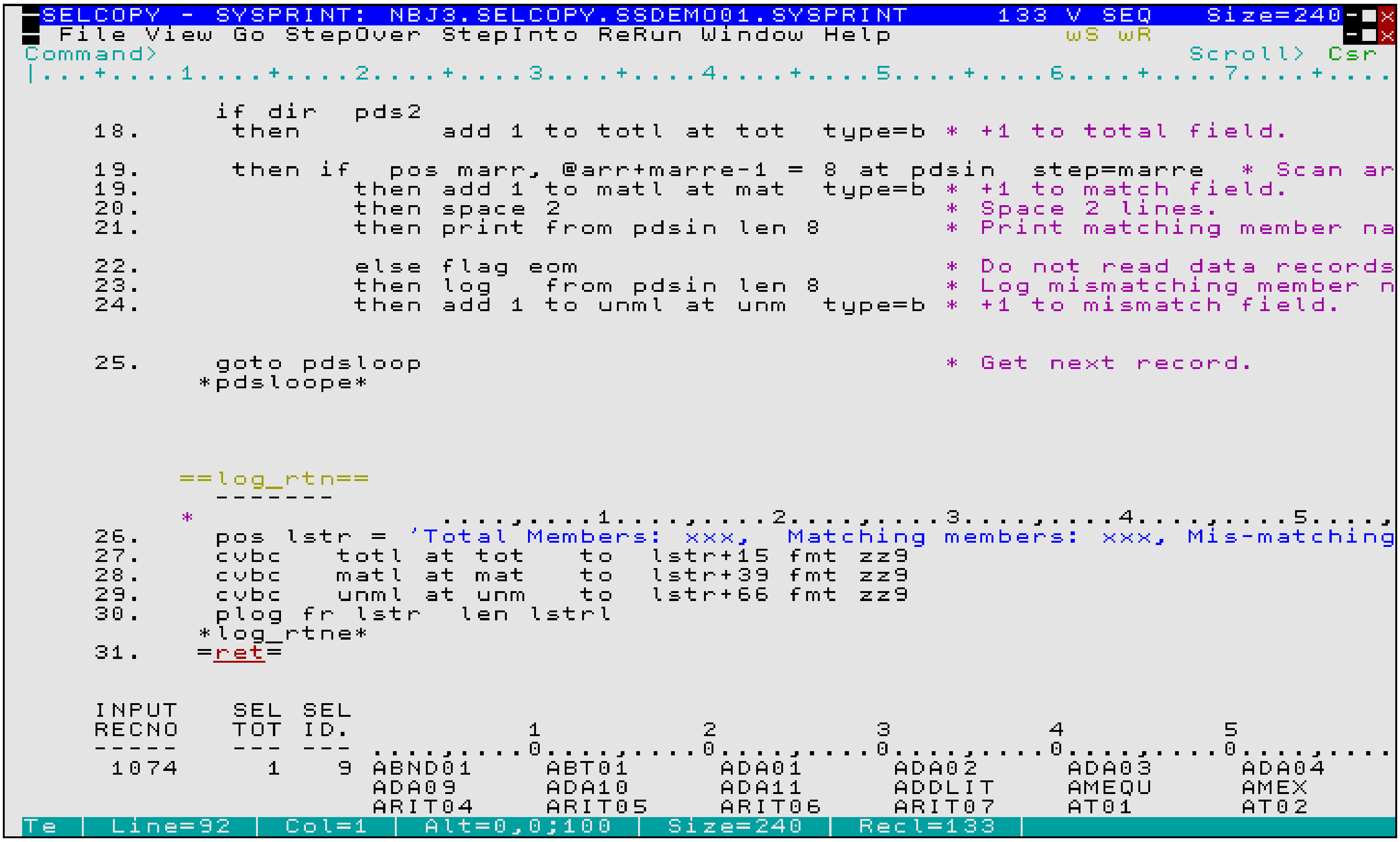
- Select 'Listing' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW LIST.

SELCOPY Debug intercepts output to SYSPRINT/SYSLST and displays it in the SYSPRINT window instead. For this reason, SYSPRINT or SYSLST does not need to be allocated and no output is written to the system spool.

The contents of the SYSPRINT window scroll automatically to display any new output to SYSPRINT/SYSLST. Data written to the SYSPRINT window is maintained until the SELCOPY Debug session is closed. Therefore, so long as the SELCOPY Debug session is not closed, the job may be re-run any number of times without loosing the SYSPRINT/SYSLST output from a previous run.

The SYSPRINT window is an edit view which supports execution of CBLe commands and macros. This allows the user to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)

Unless SELCOPY options NOPRINT or NOPCTL are specified in the control statements, the input statements and their selection ids are also written to SYSPRINT. Similarly, unless SELCOPY options NOPRINT, NOPSUM or NOPTOT are specified in the control statements, the summary totals are written to SYSPRINT at end of job.



```
 SELCOPY - SYSPRINT: NBJ3.SELCOPY.SSDEM001.SYSPRINT     133 V SEQ    Size=240
  File View Go StepOver StepInto ReRun Window Help            wS wR
Command>                                                              Scroll> Csr
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....

           if dir  pds2
    18.     then          add 1 to totl at tot   type=b * +1 to total field.

    19.     then if  pos marr, @arr+marre-1 = 8 at pdsin  step=marre  * Scan ar
    19.            then add 1 to matl at mat   type=b * +1 to match field.
    20.            then space 2                       * Space 2 lines.
    21.            then print from pdsin len 8        * Print matching member na

    22.            else flag eom                      * Do not read data records
    23.            then log   from pdsin len 8        * Log mismatching member n
    24.            then add 1 to unml at unm   type=b * +1 to mismatch field.


    25.    goto pdsloop                               * Get next record.
          *pdsloope*



         ==log_rtn==
           --------
         *              ....,....1....,....2....,....3....,....4....,....5....,
    26.    pos lstr = 'Total Members: xxx,  Matching members: xxx, Mis-matching
    27.    cvbc   totl at tot   to  lstr+15 fmt zz9
    28.    cvbc   matl at mat   to  lstr+39 fmt zz9
    29.    cvbc   unml at unm   to  lstr+66 fmt zz9
    30.    plog fr lstr  len lstrl
          *log_rtne*
    31.   =ret=


    INPUT    SEL SEL
    RECNO    TOT ID.          1         2         3         4         5
    -----    --- --- ....,....0....,....0....,....0....,....0....,....0....,....
     1074      1   9 ABND01    ABT01     ADA01     ADA02     ADA03     ADA04
                     ADA09     ADA10     ADA11     ADDLIT    AMEQU     AMEX
                     ARIT04    ARIT05    ARIT06    ARIT07    AT01      AT02
 Te | Line=92 | Col=1 | Alt=0,0;100 | Size=240 | Recl=133 |
```

Figure 64. SELCOPY SYSPRINT Window.

SQL Log Window

The SQL log window may be opened via the following:

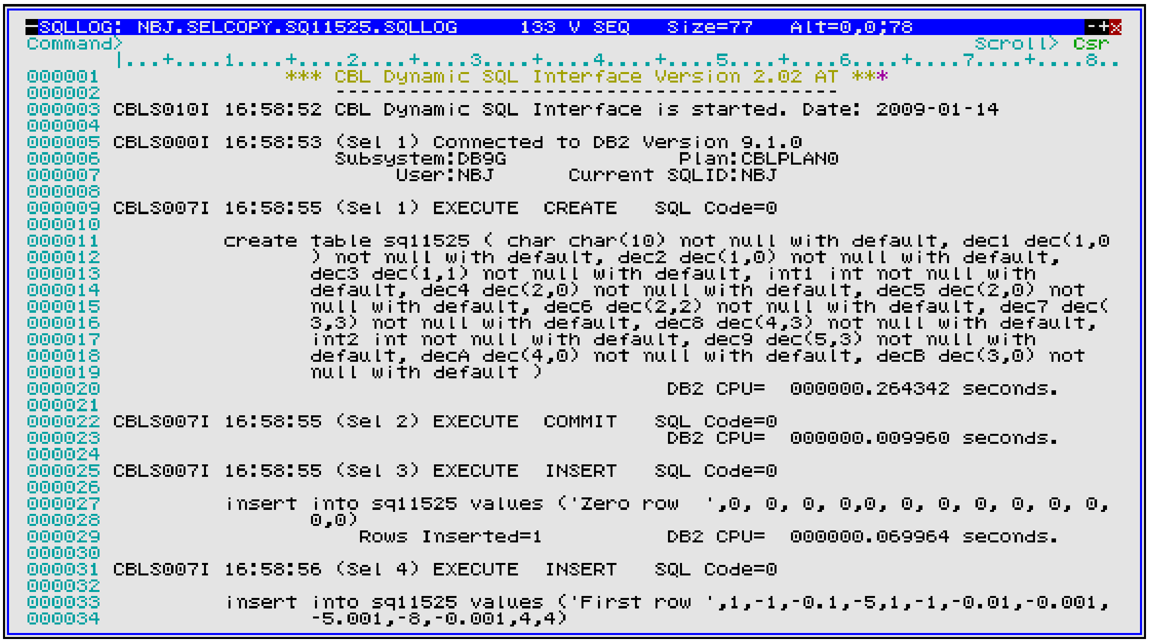
- Select 'SQL log' from the View menu in the SELCOPY Debug Main Menu.

- Enter the SELCOPY Debug CLI command WINDOW SQL.

A SELCOPY job that submits SQL statements to a DB2 data base, also writes detailed information about the SELCOPY SQL processing to a data set allocated to ddname **CBLSQLOG**.

SELCOPY Debug intercepts output to CBLSQLOG and displays it in the SQL Log window instead. Because of this, CBLSQLOG does not need to be allocated to display this information.

The SQL Log window is an edit view which supports execution of CBLe commands and macros. This allows the user to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)



```
-SQLLOG: NBJ.SELCOPY.SQ11525.SQLLOG      133 V SEQ    Size=77    Alt=0,0;78     -+x
Command>                                                                  Scroll> Csr
       |...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8..
000001               *** CBL Dynamic SQL Interface Version 2.02 AT ***
000002                   ---------------------------------------
000003 CBLS010I 16:58:52 CBL Dynamic SQL Interface is started. Date: 2009-01-14
000004
000005 CBLS000I 16:58:53 (Sel 1) Connected to DB2 Version 9.1.0
000006                   Subsystem:DB9G              Plan:CBLPLAN0
000007                        User:NBJ      Current SQLID:NBJ
000008
000009 CBLS007I 16:58:55 (Sel 1) EXECUTE  CREATE   SQL Code=0
000010
000011          create table sq11525 ( char char(10) not null with default, dec1 dec(1,0
000012                 ) not null with default, dec2 dec(1,0) not null with default,
000013                 dec3 dec(1,1) not null with default, int1 int not null with
000014                 default, dec4 dec(2,0) not null with default, dec5 dec(2,0) not
000015                 null with default, dec6 dec(2,2) not null with default, dec7 dec(
000016                 3,3) not null with default, dec8 dec(4,3) not null with default,
000017                 int2 int not null with default, dec9 dec(5,3) not null with
000018                 default, decA dec(4,0) not null with default, decB dec(3,0) not
000019                 null with default )
000020                                              DB2 CPU=  000000.264342 seconds.
000021
000022 CBLS007I 16:58:55 (Sel 2) EXECUTE  COMMIT   SQL Code=0
000023                                              DB2 CPU=  000000.009960 seconds.
000024
000025 CBLS007I 16:58:55 (Sel 3) EXECUTE  INSERT   SQL Code=0
000026
000027          insert into sq11525 values ('Zero row  ',0, 0, 0, 0,0, 0, 0, 0, 0, 0, 0,
000028                 0,0)
000029                     Rows Inserted=1          DB2 CPU=  000000.069964 seconds.
000030
000031 CBLS007I 16:58:56 (Sel 4) EXECUTE  INSERT   SQL Code=0
000032
000033          insert into sq11525 values ('First row ',1,-1,-0.1,-5,1,-1,-0.01,-0.001,
000034                 -5.001,-8,-0.001,4,4)
```

Figure 65. SELCOPY SQL Log Window.

WTO Log Window

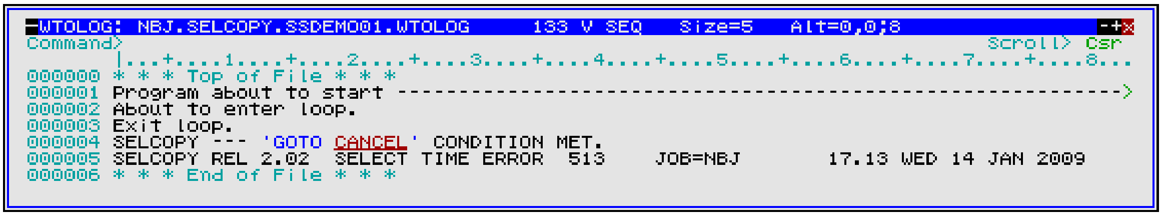
The WTO log window may be opened via the following:

- Select 'WTO log' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW WTO.

SYSLOG output to the Operator's Console, TSO, CMS or ICCF user terminals is intercepted by SELCOPY Debug and is displayed in the WTOLOG window instead.

The WTO Log window is opened automatically when SYSLOG output is received. This may be warning/error messages returned by SELCOPY, or output generated by a SELCOPY LOG operation.

The WTO Log window is an edit view which supports execution of CBLe commands and macros. This allows the user to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)



```
-WTOLOG: NBJ.SELCOPY.SSDEMO01.WTOLOG      133 V SEQ    Size=5    Alt=0,0;8      -+x
Command>                                                                  Scroll> Csr
       |...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8...
000000 * * * Top of File * * *
000001 Program about to start ----------------------------------------------------------->
000002 About to enter loop.
000003 Exit loop.
000004 SELCOPY --- 'GOTO CANCEL' CONDITION MET.
000005 SELCOPY REL 2.02  SELECT TIME ERROR  513    JOB=NBJ       17.13 WED 14 JAN 2009
000006 * * * End of File * * *
```

Figure 66. SELCOPY WTO Log Window.

Work Area/Current Input Record Window

A Work Area/Current Input Record storage display window is opened automatically when SELCOPY Debug is started. Further storage display windows may also be opened via the following:

- Select 'Work area' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW WORKAREA.

The current status of the user work area (or input record buffer if no work area is allocated) is displayed in the Work Area window. A Work area window is a storage display window.

Note that, if WORKLEN is not supplied, the Work Area window has the title: Current Input Record.

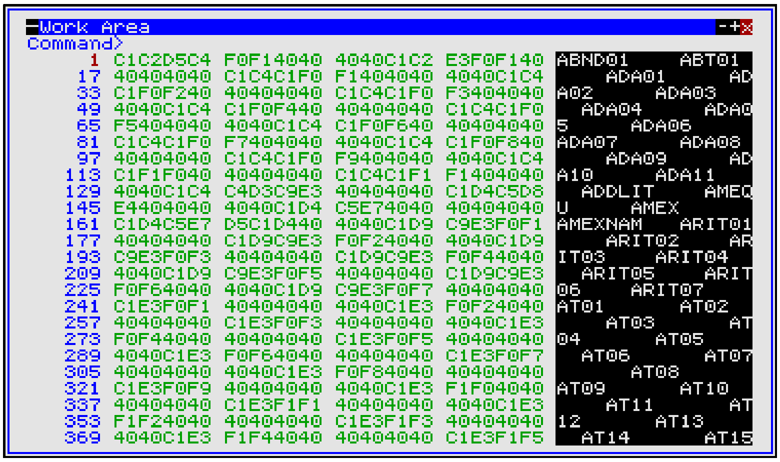
Any number of Work Area windows may be opened and each window may be tailored to display different portions of the work area.

The appearance of the Work Area window may be updated using the storage window display options popup menu. The options available and methods used to display this menu are documented under the line command, SHOWPOPUPMENU.

The work area position, in the first row of the Work Area window, is an enterable field (highlighted in red by default.) Here, you may enter the work area position from which data is to be displayed.

Line commands UP CURSOR and DOWN CURSOR may also be used to navigate the Work Area window. By default, UP CURSOR is assigned to PF07 and DOWN CURSOR is assigned to PF08.

Data in the work area may be altered at any point during the run by overtyping text in either the character or hexadecimal display. A change to text in the one display will automatically be reflected in the other.



```
=Work Area                                                    -+x
Command>
      1 C1C2D5C4 F0F14040 4040C1C2 E3F0F140 ABND01    ABT01
     17 40404040 C1C4C1F0 F1404040 4040C1C4     ADA01     AD
     33 C1F0F240 40404040 C1C4C1F0 F3404040 A02     ADA03
     49 4040C1C4 C1F0F440 40404040 C1C4C1F0   ADA04     ADA0
     65 F5404040 4040C1C4 C1F0F640 40404040 5     ADA06
     81 C1C4C1F0 F7404040 4040C1C4 C1F0F840 ADA07     ADA08
     97 40404040 C1C4C1F0 F9404040 4040C1C4     ADA09     AD
    113 C1F1F040 40404040 C1C4C1F1 F1404040 A10     ADA11
    129 4040C1C4 C4D3C9E3 40404040 C1D4C5D8   ADDLIT    AMEQ
    145 E4404040 4040C1D4 C5E74040 40404040 U     AMEX
    161 C1D4C5E7 D5C1D440 4040C1D9 C9E3F0F1 AMEXNAM   ARIT01
    177 40404040 C1D9C9E3 F0F24040 4040C1D9     ARIT02    AR
    193 C9E3F0F3 40404040 C1D9C9E3 F0F44040 IT03    ARIT04
    209 4040C1D9 C9E3F0F5 40404040 C1D9C9E3   ARIT05    ARIT
    225 F0F64040 4040C1D9 C9E3F0F7 40404040 06    ARIT07
    241 C1E3F0F1 40404040 4040C1E3 F0F24040 AT01      AT02
    257 40404040 C1E3F0F3 40404040 4040C1E3     AT03      AT
    273 F0F44040 40404040 C1E3F0F5 40404040 04      AT05
    289 4040C1E3 F0F64040 40404040 C1E3F0F7   AT06      AT07
    305 40404040 4040C1E3 F0F84040 40404040       AT08
    321 C1E3F0F9 40404040 4040C1E3 F1F04040 AT09      AT10
    337 40404040 C1E3F1F1 40404040 4040C1E3     AT11      AT
    353 F1F24040 40404040 C1E3F1F3 40404040 12      AT13
    369 4040C1E3 F1F44040 40404040 C1E3F1F5   AT14      AT15
```

Figure 67. SELCOPY Work Area Window.

POS Expression Window

The POS expression window may be opened using the SELCOPY Debug CLI command WINDOW POS expr. POS expression windows for special positions POS PARM, DATE, SQLCA, SQLDA and SQLMA may be opened by selecting the "Pos" sub-menu from the View menu in the SELCOPY Debug Main Menu.

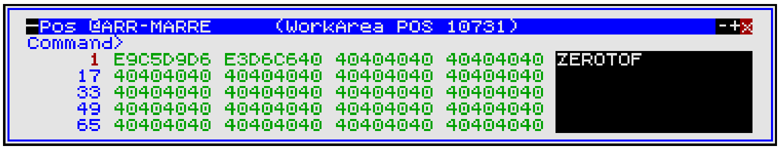
The POS window displays storage in the exactly same way as the Work Area/Current Input Record window with the exception that the start address of the displayed data is a position in storage evaluated by a valid SELCOPY POS expression instead of position 1 of the work area.

Like the Work Area window, the appearance of the POS expression window may be updated using the storage display window options popup menu. The options available and methods used to display this menu are documented under the line command, SHOWPOPUPMENU.

The POS expression is re-evaluated at each break in the SELCOPY execution and the data at the new position displayed in the POS window.

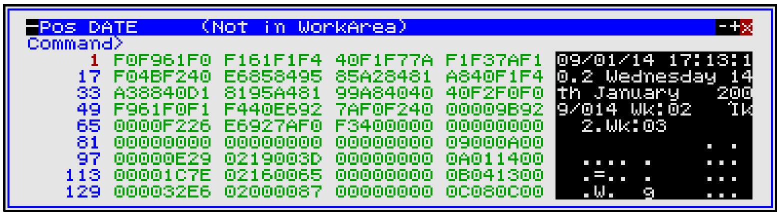
The POS window title contains the POS expression and the evaluated position in the work area in parentheses. If the evaluated position falls outside the work area, then **(Not in WorkArea)** is displayed instead.

Any number of POS windows may be opened.



```
=Pos @ARR-MARRE       (WorkArea POS 10731)                    -+x
Command>
      1 E9C5D9D6 E3D6C640 40404040 40404040 ZEROTOF
     17 40404040 40404040 40404040 40404040
     33 40404040 40404040 40404040 40404040
     49 40404040 40404040 40404040 40404040
     65 40404040 40404040 40404040 40404040
```

Figure 68. POS Window (inside work area)



```
Pos DATE       (Not in WorkArea)                                    -+x
Command>
      1 F0F961F0 F161F1F4 40F1F77A F1F37AF1 09/01/14 17:13:1
     17 F04BF240 E6858495 85A28481 A840F1F4 0.2 Wednesday 14
     33 A38840D1 8195A481 99A84040 40F2F0F0 th January   200
     49 F961F0F1 F440E692 7AF0F240 00009B92 9/014 Wk:02   Ik
     65 0000F226 E6927AF0 F3400000 00000000   2.Wk:03
     81 00000000 00000000 00000000 09000A00              . .
     97 00000E29 0219003D 00000000 0A011400   .... .     ...
    113 00001C7E 02160065 00000000 0B041300   .=.. .     ...
    129 000032E6 02000087 00000000 0C080C00   .W.   9    ...
```

Figure 69. POS Window (outside work area)

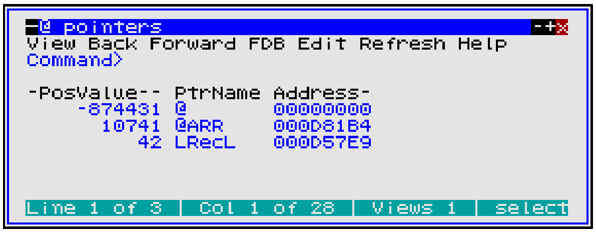
@ Pointer Window

The @ Pointer window may be opened via the following:

- Select '@ Pointers' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW @.

The current status of the @ pointer, LRECL and of all the user @ pointers to be used in the current execution of SELCOPY, is displayed in the @ Pointer window.

The @ Pointer window has the same characteristics as a SELCOPY/i List window including selecting, sorting and filtering of row and column data and "point and shoot" sorting on column headers.



```
-@ pointers                                       -+x
View Back Forward FDB Edit Refresh Help
Command>

-PosValue-- PtrName Address-
    -874431 @       00000000
      10741 @ARR    000D81B4
         42 LRecL   000D57E9




Line 1 of 3 | Col 1 of 28 | Views 1 | select
```

Figure 70. SELCOPY @ Pointer Window.

Columns Displayed

Name	Type	Description
PosValue	Int	Value as a position in the work area
PtrName	Char	Pointer Name
Address	Hex	Address in storage of position in work area

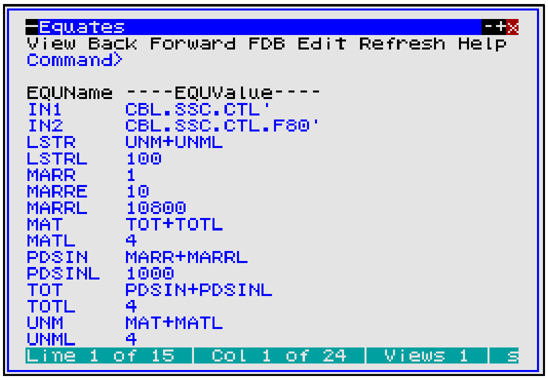
Equates Window

The Equates window may be opened via the following:

- Select 'EQUates' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW EQUATES.

All equated names and their values, set by the user via an EQU statement and subsequently allocated by SELCOPY during control statement analysis, are is displayed in the Equates window.

The Equate window has the same characteristics as a SELCOPY/i List window including selecting, sorting and filtering of row and column data and "point and shoot" sorting on column headers.



```
Equates                                    -+x
View Back Forward FDB Edit Refresh Help
Command>

EQUName  ----EQUValue----
IN1      CBL.SSC.CTL'
IN2      CBL.SSC.CTL.F80'
LSTR     UNM+UNML
LSTRL    100
MARR     1
MARRE    10
MARRL    10800
MAT      TOT+TOTL
MATL     4
PDSIN    MARR+MARRL
PDSINL   1000
TOT      PDSIN+PDSINL
TOTL     4
UNM      MAT+MATL
UNML     4
Line 1 of 15 | Col 1 of 24 | Views 1 | S
```

Figure 71. SELCOPY EQUates Window.

Columns Displayed

Name	Type	Description
EQUName	Char	Equated name
EQUValue	Char	Equated value

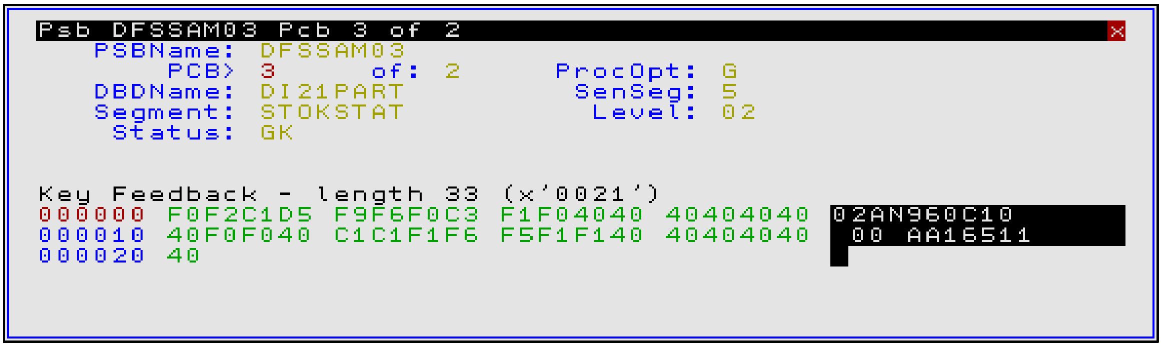
PCB Window

The PCB window may be opened via the following:

- Select 'PCB' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW PCB.

This window shows the PCB which was used to execute the most recent IMS call

The PCB displayed will change if different PCBs are used in the SELCOPY program.



```
Psb DFSSAM03 Pcb 3 of 2                                      x
    PSBName: DFSSAM03
        PCB> 3      of:  2      ProcOpt: G
    DBDName: DI21PART             SenSeg: 5
    Segment: STOKSTAT              Level: 02
     Status: GK


Key Feedback - length 33 (x'0021')
000000 F0F2C1D5 F9F6F0C3 F1F04040 40404040  02AN960C10
000010 40F0F040 C1C1F1F6 F5F1F140 40404040   00  AA16511
000020 40
```

Figure 72. SELCOPY PCB Window.

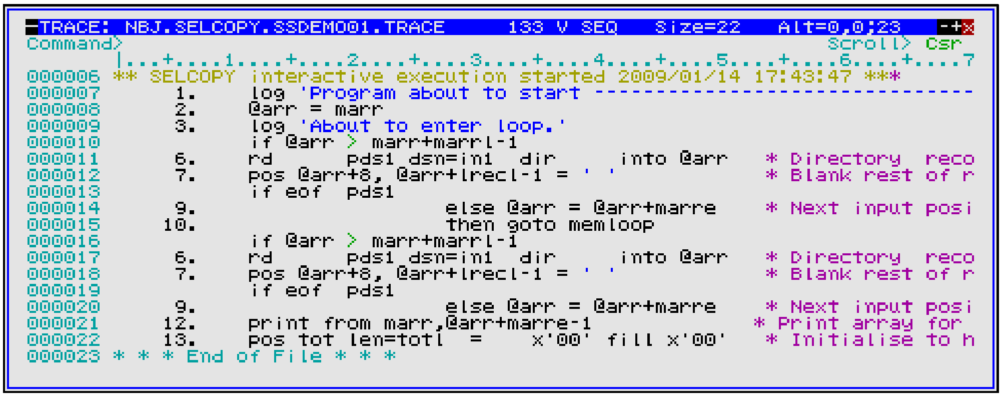
TRACE Window

The TRACE window is opened automatically when SELCOPY Debug is started. It may also be opened via the following:

- Select 'Execution trace' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW TRACE.

The TRACE window is a CBLe edit view that contains all the SELCOPY control statements at which processing has been stopped. i.e a break point was set and encountered. Each logged statement begins with the statement's selection id.

Note that the STEPINTO and STEPOVER commands dynamically set and unset break points to allow stepping through the SELCOPY job. The STEPINTO command sets a break point on the next control statement to be executed following the current control statement. Therefore, when repeatedly issuing STEPINTO or STEPOVER, the TRACE window displays a log of all the statements executed so far.



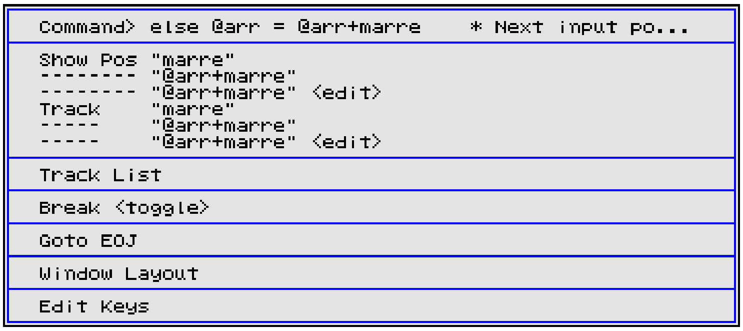
```
-TRACE: NBJ.SELCOPY.SSDEMO001.TRACE       133 V SEQ    Size=22    Alt=0,0;23   -+x
Command>                                                                Scroll> Csr
       |...+....1....+....2....+....3....+....4....+....5....+....6....+....7
000006 ** SELCOPY interactive execution started 2009/01/14 17:43:47 ***
000007       1.     log 'Program about to start ------------------------------
000008       2.     @arr = marr
000009       3.     log 'About to enter loop.'
000010              if @arr > marr+marrl-1
000011       6.     rd        pds1 dsn=in1   dir       into @arr    * Directory  reco
000012       7.     pos @arr+8, @arr+lrecl-1 = ' '                  * Blank rest of r
000013              if eof   pds1
000014       9.                    else @arr = @arr+marre           * Next input posi
000015      10.                    then goto memloop
000016              if @arr > marr+marrl-1
000017       6.     rd        pds1 dsn=in1   dir       into @arr    * Directory  reco
000018       7.     pos @arr+8, @arr+lrecl-1 = ' '                  * Blank rest of r
000019              if eof   pds1
000020       9.                    else @arr = @arr+marre           * Next input posi
000021      12.     print from marr,@arr+marre-1                   * Print array for
000022      13.     pos tot len=totl  =    x'00' fill x'00'         * Initialise to h
000023 * * * End of File * * *
```

Figure 73. SELCOPY TRACE Window.

Point-and-Shoot Popup Menu

All CBLe type SELCOPY Debug windows, including the SYSIN/SYSIPT and SYSPRINT/SYSLST windows, support the point-and-shoot options popup menu. The popup menu is opened using the SDBPOPUP edit macro defined on PF4 by default.

The cursor position within the edited data identifies the focus text to be referenced in items of the point-and-shoot menu when it is opened.



```
Command> else @arr = @arr+marre    * Next input po...

Show Pos "marre"
-------- "@arr+marre"
-------- "@arr+marre" <edit>
Track    "marre"
-----    "@arr+marre"
-----    "@arr+marre" <edit>

Track List

Break <toggle>

Goto EOJ

Window Layout

Edit Keys
```

Figure 74. SELCOPY Point-and-Shoot Popup Menu Window.

The menu enables the user to quickly and easily perform the following, commonly used tasks:

CmdText
Execute the CBLe CMDTEXT command against the command string text at the focus position.

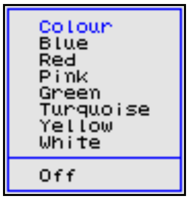
Show Pos "*expression*"
Execute the SELCOPY Debug CLI command WINDOW POS expr to open a POS storage display window starting at the position defined by expression. Show POS entries exist for POS expressions determined by the token on which focus was placed when the menu was opened. Possible expressions are:

1. The focus token that forms part of a larger positional expression.
2. If applicable, the full positional expression containing the focus token.

Show Pos "*expression*" <edit>
Same as the **Show Pos** entry but instead of executing the **WINDOW POS expr** command directly, it is placed at the edit command prompt ready for edit and submission by the user.

Track "*expression*"
Invoke the **SDBTRACK** edit macro which issues the SELCOPY Debug CLI command TRACK expr to start or stop tracking a position in storage referenced by the POS expression. Track entries exist for POS expressions determined as for "Show Pos".

If a Track entry is selected, another popup menu is opened prompting the user to select the colour to be used for tracking this POS expression or, alternatively, to turn off tracking for this POS expression.



```
Colour
Blue
Red
Pink
Green
Turquoise
Yellow
White

Off
```

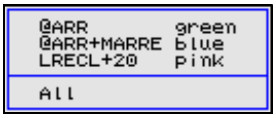
Figure 75. SELCOPY TRACK Colour Popup Menu Window.

Track "*expression*" <edit>

Same as the **Track** entry but instead the generated "SDBTRACK expression" macro invocation is placed at the edit command prompt ready for edit and submission by the user.

Track List

Open a popup menu displaying a list of all POS expressions that are being tracked. The user can then select an entry to switch off tracking for that POS expression or select **All** to switch off tracking of all the POS expressions entries.



```
@ARR          green
@ARR+MARRE    blue
LRECL+20      pink

All
```

Figure 76. SELCOPY TRACK List Popup Window.

Break <toggle>

Toggle a break point on and off for the SELCOPY operation at the focus position.

Window Layout

Invoke the **SDBWINX** edit macro which opens a popup menu enabling the user to control the configuration of windows within the SELCOPY Debug MDI environment.
The user can save and subsequently restore the characteristics of the current MDI child window or all currently open MDI child windows. Alternatively, the user can select the default configuration for the current, or all SELCOPY Debug MDI child windows.

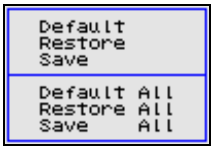


Figure 77. SELCOPY Window Layout Popup Menu.

Edit Keys/Debug Keys

Certain default PFKey assignments for CBLe text edit differ to defaults set up for the SELCOPY Debug environment. This entry enables the user to toggle between the CBLe (Edit) default keys and SELCOPY Debug (Debug) keys as follow:

PFKey	Edit	Debug
PF1	SOS LINEADD	STEPOVER
PF2	DUPLICATE	STEPINTO
PF13	SOS LINEDEL	GO
PF14	SPLTJOIN	BR ON CURSOR PERM

SELCOPY Debug Commands

You can issue SELCOPY Debug commands from the command line at the Command> prompt. Most SELCOPY Debug main window menu options have a command line equivalent.

Command	Description	PF Key
BREAKPOINT	Set/unset temporary break points.	**PF14**
EOJ	Force SELCOPY End-of-Job.	**-**
GO	Continue processing.	**PF13**
RERUN	Re-run from the beginning.	**-**
STEPINTO	Step (Trace) Into sub-routines.	**PF2**
STEPOVER	Step (Trace) Over sub-routines.	**PF1**
TRack	Track a SELCOPY POS expression.	**-**
Window	Open a specified SELCOPY Debug window.	**-**

See SELCOPY Debug Function Keys for complete list of default PFKeys.

BREAKPOINT

Syntax:

```
>>-- BReakpoint -------------------------------------------------------------><
```

Description:

Use the BREAKPOINT command to set or unset a permanent BReak point at the focus statement in the Control Card window. By default, BREAKPOINT is assigned to PF06.

If a break point is set at a particular control statement, then processing will be paused on the next attempt to execute that statement.

Any number of concurrent break points may be active during job execution.

Parameters:

BREAKPOINT has no parameters.

EOJ

Syntax:

```
>>-- EOJ --------------------------------------------------------------------><
```

Description:

Use the EOJ command to force SELCOPY to immediately execute a "GOTO EOJ" operation.

The SELCOPY job will end without processing any further control statements and will generate its output summary in the SYSPRINT window.

Parameters:

EOJ has no parameters.

GO

Syntax:

```
>>-- GO ---------------------------------------------------------------------><
```

Description:

Use the GO command to continue processing of the control statements. By default, GO is assigned to PF04.

Processing will continue until a break point or End-of-Job is encountered at which point processing is paused or stopped respectively.

Parameters:

GO has no parameters.

RERUN

Syntax:

```
>>--+- RErun --+-------------------------------------------------------------><
    |          |
    +- RR -----+
```

Description:

Use the RERUN command to Re-Run the job from the beginning. No further statements will be executed from the existing job run.

Storage is cleared and values and break points, set during control statement analysis, are re-initialised. All existing breakpoints are cleared.

Parameters:

RERUN has no parameters.

STEPINTO

Syntax:

```
>>--+- STEPInto --+-----------------------------------------------------------><
    |             |
    +- SI --------+
```

Description:

Use the STEPINTO command to step through the SELCOPY control statements logically one at a time. By default, STEPINTO is assigned to PF02.

Any branch to a SELCOPY sub-routine via a **DO**, **PERFORM** or **GOSUB** operation will be Stepped Into. i.e. processing is paused on each control statement in the sub-routine.

STEPINTO and STEPOVER set and then unset temporary break points in the SELCOPY control statements in order to pause processing.

Parameters:

STEPINTO has no parameters.

STEPOVER

Syntax:

```
>>--+- STEPOver --+-----------------------------------------------------------><
    |             |
    +- SO --------+
```

Description:

Use the STEPOVER command to step through the SELCOPY control statements logically one at a time. By default, STEPOVER is assigned to PF01.

Any branch to a SELCOPY sub-routine via a **DO**, **PERFORM** or **GOSUB** operation will be Stepped Over. i.e. the sub-routine is executed and processing is paused again on the control statement following the sub-routine call.

STEPINTO and STEPOVER set and then unset temporary break points in the SELCOPY control statements in order to pause processing.

Parameters:

STEPOVER has no parameters.

TRACK

Syntax:

```
>>-- TRack -- expr ---+--------------+----------------------------------------><
                      |              |
                      +--- colour ---+
                      |              |
                      +---- OFF -----+
```

Description:

Use the TRACK command to track the value of a valid SELCOPY POS expression as a position in storage.

The single byte, addressed by the POS expression, is highlighted in all open storage windows in which the position is displayed.

The POS expression is re-evaluated for every break in the SELCOPY execution.

Parameters:

`expr`

A valid SELCOPY POS expression. This may include EQUated names, @ pointers, LRECL special POS keywords (e.g. DATE, COMREG), integer values and arithmetic operators "+" (plus) and "-" (minus).

`colour`

The colour in which the evaluated position is highlighted. This is a two character code defining the colour and, optionally the extended highlighting, to be used.
Valid colour codes are:

B	Blue
G	Green
P	Pink
R	Red
T	Turquoise
W	White
Y	Yellow

Valid extended highlighting codes are:

B	Blink
N	None (No extended highlighting)
R	Reverse Video
U	Underscore

The default extended highlighting is **R** (reverse video), the default colour is **T** (turquoise).

`OFF`

Switch off tracking for the specified expression.

Examples:

`TRACK @A+10  R`

Highlight in red (default reverse video) the byte in all storage windows that is referenced by the expression @A+10.

`TRACK ARRAY+@X-1  GU`

Highlight in green with underscore the byte in all storage windows that is referenced by the expression ARRAY+@X-1.

WINDOW

Syntax:

```
>>-- Window --+--- @ ----------+---------------------------------------------><
              |                |
              +--- AT----------+
              |                |
              +--- Ctl --------+
              |                |
              +--- EQuates ----+
              |                |
              +--- List -------+
              |                |
              +--- PCB --------+
              |                |
              +--- POS expr ---+
              |                |
              +--- SQL --------+
              |                |
              +--- Workarea ---+
              |                |
              +--- WTO --------+
              |                |
              +--- TRace ------+
```

Description:

Use the WINDOW command to open and place focus on the nominated window type.

Windows may also be opened via the Window menu of the SELCOPY Debug main window menu bar.

Parameters:

`@`
`AT`
Open and place focus on the @ Pointer window.

`CTL`
Open and place focus on the Control Cards window.
Note that, closing the Control Cards window also exits SELCOPY Debug.

`EQUATES`
Open and place focus on the Equates window.

`LIST`
Open and place focus on the Output Listing window.
Note that, closing the Output Listing window also exits SELCOPY Debug.

`PCB`
Open the PCB window.

`POS expr`
Open a POS window. A valid SELCOPY POS expression must be specified to define the start address of the storage display.

`SQL`
Open the SQL Log window.

`WORKAREA`
Open a storage window (Work Area window.)

`WTO`
Open the WTO Log window.

`TRACE`
Open and place focus on the Trace window.
Note that, closing the Trace window also exits SELCOPY Debug.

SELCOPY Debug Function Keys

You can assign 3270 Program Function Keys (PFKeys) to line commands as applicable to the window class. The KEYS line command may be used to display and globally assign Function key settings.

SELCOPY Debug PFKeys have default functions assigned as determined by the window's class (i.e. Edit, Storage or List.) In addition to these, edit views and storage windows have the following PFKey definitions:

PF1	StepOver	Execute the next SELCOPY operation (step over a sub-routine.)
PF2	StepInto	Execute the next SELCOPY operation (step into a sub-routine.)
PF4	Macro SdbPopup	Invoke the SDBPOPUP edit macro to display a functions menu (edit views only.)
PF5	ShowPopupMenu	Open the storage display options popup menu (storage display windows only.)
PF13	Go	Continue SELCOPY processing.
PF14	BreakPoint	Toggle a break point on and off at the focus operation (edit views only.)

For edit views only, the default action of the PF1/PF2/PF13/PF14 keys may be governed via the SdbPopup macro which allows the user to select either "Edit" or "Debug" PFKey configuration.

Utilities Menu (=8)

The Utilities Menu panel (ZZSGUTIL) is an interactive panel window opened on selection of option 8. in the SELCOPY/i Primary option menu.

SELCOPY/i supports a number of general purpose utilities and interfaces to a selection of system utilities that may be accessed via this panel.

Options

1	SELCOPY/debug	SELC - SELCOPY/batch language interactive debug
2	CBLVCAT	VCAT - Catalog/VTOC report online excution
3	IDCAMS	AMS - Execute IDCAMS commands interactively
4	Catalog ALIAS	AMSA - Define new Catalog Alias
5	Library ALIAS	ALI - Create new PDS/PDSE library member Alias
6	IEBCOPY	IEBC - Execute IEBCOPY interactively
7	Favourites	FAV - Favourite Datesets/Commands
8	System	SY - Display System Information
9	Search	FS - Basic PDS/PDSE Library string search
10	Find Lib Mbr(s)	LLX - Search for member(s) across multiple libraries
11	Compare Files	COMPF - Compare Files
12	Compare Libs	COMPL - Compare Libraries
13	Calendar	CAL - Basic Calendar
14	Calculator	CALC - REXX expression calculator

CBLVCAT Interactive (VCI) (=8.2)

If a valid software licence key has been applied, SELCOPY/i may invoke the CBLVCAT program to allow interactive execution within a SELCOPY/i window.

1. CBLVCAT Interactive Window
2. Raw Data Window

CBLVCAT Interactive Window

The Execute CBLVCAT window is used to execute CBLVCAT Interactive and may be opened via the following:

- Select option 2. 'CBLVCAT' from the Utilities Menu
- Select 'CBLVCAT Interactive' from the File menu in the CBLe main window menu bar.
- Enter the command VCAT on the command line of any window.
- Enter the "T" or "VC" prefix command in the prefix area of an existing Execute CBLVCAT window or certain other List type windows. "T" will generate a CBLVCAT Tune report and IDCAMS DEFINE deck for a VSAM file, "VC" will generate a CBLVCAT catalog and/or VTOC report for the list entry.

CBLVCAT is used to generate standard and customised reports on VTOC and ICF/VSAM catalog data. It also supports VSAM file tuning and generation of IDCAMS DEFINE job source.
Details on CBLVCAT output and control statement syntax is found in the CBLVCAT User Manual.

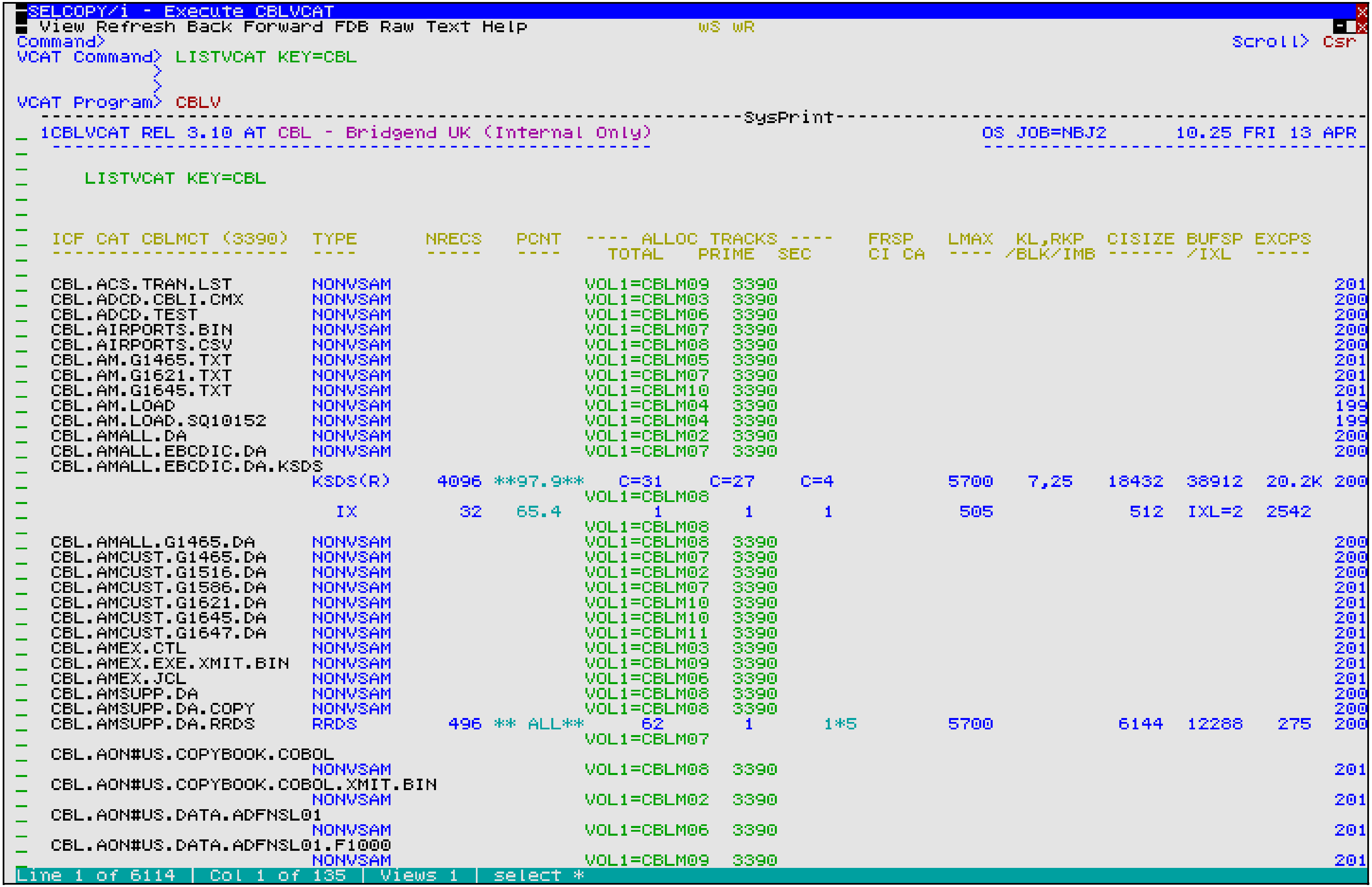
SELCOPY/i loads CBLVCAT and assumes control over its control statement input and report output functions. This allows the user to specify CBLVCAT input statements directly at the VCAT Command prompt or indirectly via a control statement file and view the output in a window.

In order to direct input from a control statement file, the fileid should be entered at the VCAT Command prompt and prefixed with a "<" (less than) symbol.

If you are using the LISTVCAT DEFINE option then the generated IDCAMS control statements are displayed in a CBLe text edit window and may subsequently be saved to a file.

After execution of CBLVCAT control statements (or control statement file), the SYSPRINT (MVS and CMS) or SYSLST (VSE) output is presented in the display area of the Execute CBLVCAT window.

The Execute CBLVCAT window display area is a list window with a single column (i.e. SysPrint) and so has charactersitics defined by the list window class. For example, the Execute CBLVCAT window supports Prefix Commands and filtering, to display new views of the data.

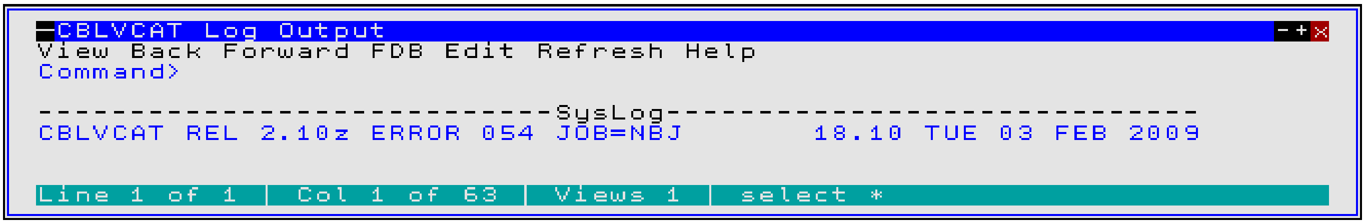


```
 SELCOPY/i - Execute CBLVCAT
  View Refresh Back Forward FDB Raw Text Help                 wS wR
Command>                                                                                                   Scroll> Csr
VCAT Command> LISTVCAT KEY=CBL
             >
             >
VCAT Program> CBLV
  --------------------------------------------------------------SysPrint--------------------------------------------------------
_ 1CBLVCAT REL 3.10 AT CBL - Bridgend UK (Internal Only)                             OS JOB=NBJ2      10.25 FRI 13 APR
_  -----------------------------------------------------                              ----------------------------------
_
_     LISTVCAT KEY=CBL
_
_
_
_  ICF CAT CBLMCT (3390)  TYPE      NRECS   PCNT  ---- ALLOC TRACKS ----   FRSP   LMAX  KL,RKP  CISIZE BUFSP EXCPS
_  ---------------------  ----      -----   ----    TOTAL   PRIME  SEC     CI CA  ---- /BLK/IMB ------ /IXL  -----
_
_  CBL.ACS.TRAN.LST       NONVSAM                 VOL1=CBLM09  3390                                                 201
_  CBL.ADCD.CBLI.CMX      NONVSAM                 VOL1=CBLM03  3390                                                 200
_  CBL.ADCD.TEST          NONVSAM                 VOL1=CBLM06  3390                                                 200
_  CBL.AIRPORTS.BIN       NONVSAM                 VOL1=CBLM07  3390                                                 200
_  CBL.AIRPORTS.CSV       NONVSAM                 VOL1=CBLM08  3390                                                 200
_  CBL.AM.G1465.TXT       NONVSAM                 VOL1=CBLM05  3390                                                 201
_  CBL.AM.G1621.TXT       NONVSAM                 VOL1=CBLM07  3390                                                 201
_  CBL.AM.G1645.TXT       NONVSAM                 VOL1=CBLM10  3390                                                 201
_  CBL.AM.LOAD            NONVSAM                 VOL1=CBLM04  3390                                                 199
_  CBL.AM.LOAD.SQ10152    NONVSAM                 VOL1=CBLM04  3390                                                 199
_  CBL.AMALL.DA           NONVSAM                 VOL1=CBLM02  3390                                                 200
_  CBL.AMALL.EBCDIC.DA    NONVSAM                 VOL1=CBLM07  3390                                                 200
_  CBL.AMALL.EBCDIC.DA.KSDS
_                         KSDS(R)    4096 **97.9**   C=31    C=27    C=4          5700   7,25   18432  38912  20.2K 200
_                                                 VOL1=CBLM08
_                           IX         32   65.4        1       1      1           505          512  IXL=2  2542
_                                                 VOL1=CBLM08
_  CBL.AMALL.G1465.DA     NONVSAM                 VOL1=CBLM08  3390                                                 200
_  CBL.AMCUST.G1465.DA    NONVSAM                 VOL1=CBLM07  3390                                                 200
_  CBL.AMCUST.G1516.DA    NONVSAM                 VOL1=CBLM02  3390                                                 200
_  CBL.AMCUST.G1586.DA    NONVSAM                 VOL1=CBLM07  3390                                                 201
_  CBL.AMCUST.G1621.DA    NONVSAM                 VOL1=CBLM10  3390                                                 201
_  CBL.AMCUST.G1645.DA    NONVSAM                 VOL1=CBLM10  3390                                                 201
_  CBL.AMCUST.G1647.DA    NONVSAM                 VOL1=CBLM11  3390                                                 201
_  CBL.AMEX.CTL           NONVSAM                 VOL1=CBLM03  3390                                                 201
_  CBL.AMEX.EXE.XMIT.BIN  NONVSAM                 VOL1=CBLM09  3390                                                 201
_  CBL.AMEX.JCL           NONVSAM                 VOL1=CBLM06  3390                                                 201
_  CBL.AMSUPP.DA          NONVSAM                 VOL1=CBLM08  3390                                                 200
_  CBL.AMSUPP.DA.COPY     NONVSAM                 VOL1=CBLM08  3390                                                 200
_  CBL.AMSUPP.DA.RRDS     RRDS        496 ** ALL**    62       1      1*5         5700         6144  12288   275  200
_                                                 VOL1=CBLM07
_  CBL.AON#US.COPYBOOK.COBOL
_                         NONVSAM                 VOL1=CBLM08  3390                                                 201
_  CBL.AON#US.COPYBOOK.COBOL.XMIT.BIN
_                         NONVSAM                 VOL1=CBLM02  3390                                                 201
_  CBL.AON#US.DATA.ADFNSL01
_                         NONVSAM                 VOL1=CBLM06  3390                                                 201
_  CBL.AON#US.DATA.ADFNSL01.F1000
_                         NONVSAM                 VOL1=CBLM09  3390                                                 201
 Line 1 of 6114 | Col 1 of 135 | Views 1 | select *
```

Figure 78. CBLVCAT Interactive in 58x120 3270 Session.

In addition to the standard List window menu items, the Execute CBLVCAT window includes the menu item **RAW** to open the CBLVCAT LISTVCAT or LISTVTOC Raw Data window.

CBLVCAT Log Output window is opened only if the CBLVCAT execution has generated SYSLOG output. This usually occurs if an error has been encountered in which case an information window is also displayed.


CBLVCAT Log Output
View Back Forward FDB Edit Refresh Help
Command>

```
------------------------------SysLog----------------------------
CBLVCAT REL 2.10z ERROR 054 JOB=NBJ        18.10 TUE 03 FEB 2009
```

Line 1 of 1 | Col 1 of 63 | Views 1 | select *

Figure 79. CBLVCAT Log Output window.

`VCAT Command>`

Enter one of the following:

◊ A CBLVCAT command, as you would code it on a CBLVCAT control statement.
◊ <filename, where filename is the name of a CBLVCAT control statement file.

The CBLVCAT command syntax is described in the CBLVCAT User Manual.

`VCAT Program>`

Specify the name of the CBLVCAT executable MODULE (MVS) or PHASE (VSE). By default, this field contains CBLV.

Prefix Commands

The following prefix area commands are available:

Command	Description
<Dflt>	See **Note 1** below.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
F	Open the FSU - File Search/Update Window to perform advanced file search and optionally update.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS/PDSE, open the file search window for the PDS.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry. (default)
K	Delete (Kill) the entry without prompting for verification.
L	Open a Dataset List window for the entry.
M	If the entry is a PDS/PDSE, open a Library List window.
Q	Open a Dataset Enqueue List window for the entry (major name SYSDSN.)
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open another Execute CBLVCAT window and issue a LISTVCAT with TUNE DEFINE to generate tuned output for the entry.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open another Execute CBLVCAT window and issue a LISTVCAT and/or LISTVTOC operation (as appropriate) for the entry. See **Note 2**.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry. Note that this command will only be successful for lines of a LISTVCAT report containing VOLn=volser.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Notes:

1. The default action on hitting <Enter> or, if configured, double-clicking the left mouse button on a SysPrint line depends on the contents of the report entry, as follows:
   1. If the entry contains the TYPE field "USERCAT" or "ALIAS OF", then prefix command "VC" is default.
   2. If the entry contains the TYPE field "PDS" or "PDSE", then prefix command "M" (Member List) is default.
   3. If the entry contains a fileid, then prefix command "E" (Edit) is default.

2. The "VC" prefix command performs LISTVCAT/VTOC operations based on the contents of the entry fields, as follow:
   1. If the entry contains the TYPE field "USERCAT", then a new report is generated for the entire contents of that catalog. Otherwise, only list the catalog entries that match the fileid.
   2. If the report entry also contains a "VOLn=volser" field, then generate a LISTVTOC report for entries that match the fileid in the volume's VTOC.

Columns Displayed

Name	Type	Description
SysPrint	Char	VCAT output report line

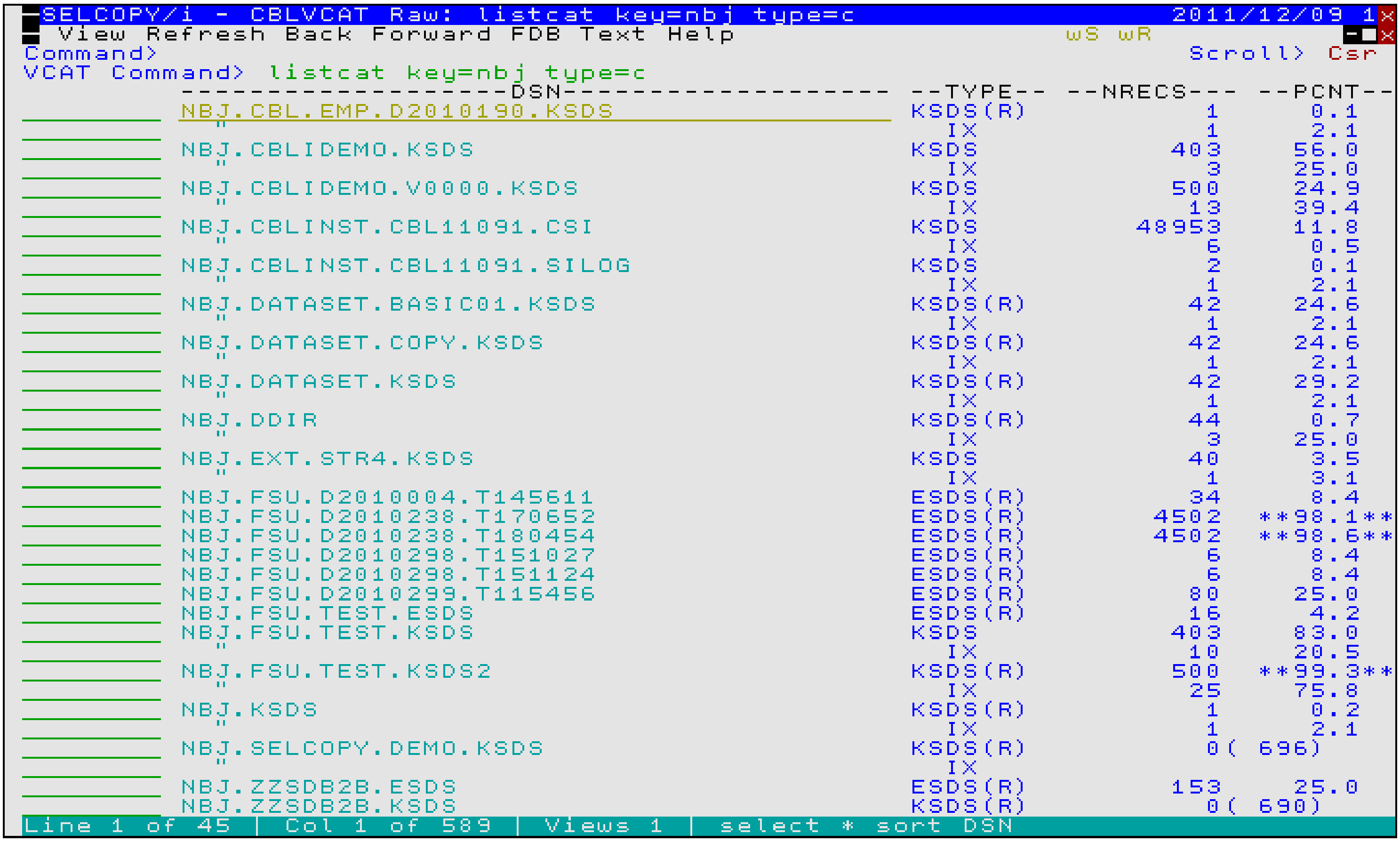
Raw Data Window

The CBLVCAT Raw window may be opened via the following:

- Select 'Raw' from the menu item of the Execute CBLVCAT window.
- Enter the command LVR on the command line of any window.

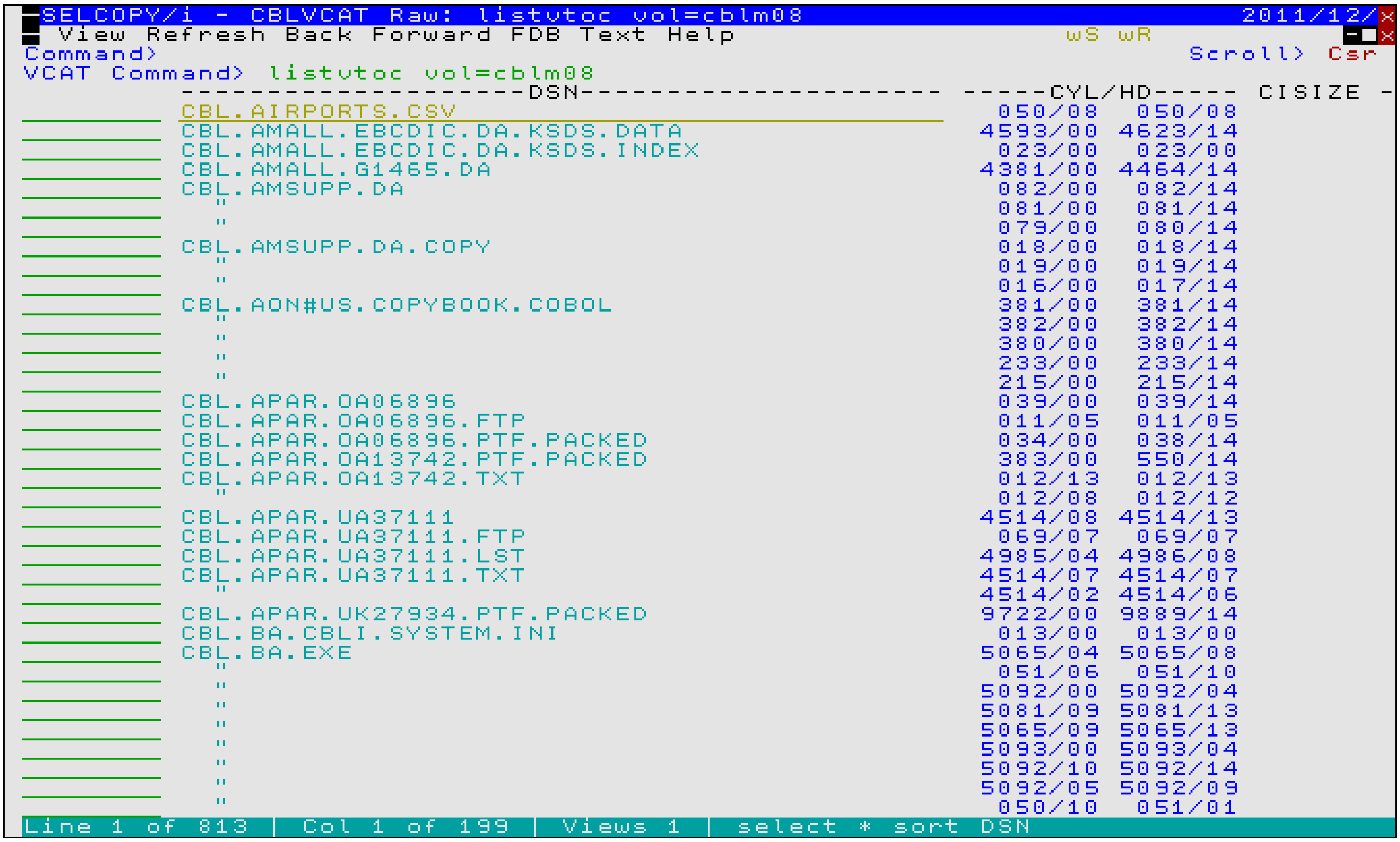
Where CBLVCAT arranges data in a printable report format, the CBLV Raw Data window provides a list of all report field data accumulated by CBLVCAT in order to generate the report.

The CBLVCAT Raw window has the same characteristics as a SELCOPY/i List window including selecting, sorting and filtering of row and column data and "point and shoot" sorting on column headers.



```
 SELCOPY/i - CBLVCAT Raw: listcat key=nbj type=c                 2011/12/09 1
  View Refresh Back Forward FDB Text Help                          wS wR
Command>                                                          Scroll> Csr
VCAT Command> listcat key=nbj type=c
         -------------------DSN--------------------  --TYPE--  --NRECS---  --PCNT--
         NBJ.CBL.EMP.D2010190.KSDS                 KSDS(R)          1     0.1
           "                                         IX             1     2.1
         NBJ.CBLIDEMO.KSDS                         KSDS           403    56.0
           "                                         IX             3    25.0
         NBJ.CBLIDEMO.V0000.KSDS                   KSDS           500    24.9
           "                                         IX            13    39.4
         NBJ.CBLINST.CBL11091.CSI                  KSDS         48953    11.8
           "                                         IX             6     0.5
         NBJ.CBLINST.CBL11091.SILOG                KSDS             2     0.1
           "                                         IX             1     2.1
         NBJ.DATASET.BASIC01.KSDS                  KSDS(R)         42    24.6
           "                                         IX             1     2.1
         NBJ.DATASET.COPY.KSDS                     KSDS(R)         42    24.6
           "                                         IX             1     2.1
         NBJ.DATASET.KSDS                          KSDS(R)         42    29.2
           "                                         IX             1     2.1
         NBJ.DDIR                                  KSDS(R)         44     0.7
           "                                         IX             3    25.0
         NBJ.EXT.STR4.KSDS                         KSDS            40     3.5
           "                                         IX             1     3.1
         NBJ.FSU.D2010004.T145611                  ESDS(R)         34     8.4
         NBJ.FSU.D2010238.T170652                  ESDS(R)       4502  **98.1**
         NBJ.FSU.D2010238.T180454                  ESDS(R)       4502  **98.6**
         NBJ.FSU.D2010298.T151027                  ESDS(R)          6     8.4
         NBJ.FSU.D2010298.T151124                  ESDS(R)          6     8.4
         NBJ.FSU.D2010299.T115456                  ESDS(R)         80    25.0
         NBJ.FSU.TEST.ESDS                         ESDS(R)         16     4.2
         NBJ.FSU.TEST.KSDS                         KSDS           403    83.0
           "                                         IX            10    20.5
         NBJ.FSU.TEST.KSDS2                        KSDS(R)        500  **99.3**
           "                                         IX            25    75.8
         NBJ.KSDS                                  KSDS(R)          1     0.2
           "                                         IX             1     2.1
         NBJ.SELCOPY.DEMO.KSDS                     KSDS(R)          0(  696)
           "                                         IX
         NBJ.ZZSDB2B.ESDS                          ESDS(R)        153    25.0
         NBJ.ZZSDB2B.KSDS                          KSDS(R)          0(  690)
Line 1 of 45 | Col 1 of 589 | Views 1 | select * sort DSN
```

Figure 80. CBLVCAT LISTVCAT Raw Data Window.



```
 SELCOPY/i - CBLVCAT Raw: listvtoc vol=cblm08                            2011/12/
   View Refresh Back Forward FDB Text Help                          wS wR
Command>                                                              Scroll> Csr
VCAT Command> listvtoc vol=cblm08
         --------------------DSN---------------------  -----CYL/HD-----  CISIZE -
________ CBL.AIRPORTS.CSV                                050/08   050/08
________ CBL.AMALL.EBCDIC.DA.KSDS.DATA                  4593/00  4623/14
________ CBL.AMALL.EBCDIC.DA.KSDS.INDEX                  023/00   023/00
________ CBL.AMALL.G1465.DA                             4381/00  4464/14
________ CBL.AMSUPP.DA                                   082/00   082/14
________    "                                            081/00   081/14
________    "                                            079/00   080/14
________ CBL.AMSUPP.DA.COPY                              018/00   018/14
________    "                                            019/00   019/14
________    "                                            016/00   017/14
________ CBL.AON#US.COPYBOOK.COBOL                       381/00   381/14
________    "                                            382/00   382/14
________    "                                            380/00   380/14
________    "                                            233/00   233/14
________    "                                            215/00   215/14
________ CBL.APAR.OA06896                                039/00   039/14
________ CBL.APAR.OA06896.FTP                            011/05   011/05
________ CBL.APAR.OA06896.PTF.PACKED                     034/00   038/14
________ CBL.APAR.OA13742.PTF.PACKED                     383/00   550/14
________ CBL.APAR.OA13742.TXT                            012/13   012/13
________    "                                            012/08   012/12
________ CBL.APAR.UA37111                               4514/08  4514/13
________ CBL.APAR.UA37111.FTP                            069/07   069/07
________ CBL.APAR.UA37111.LST                           4985/04  4986/08
________ CBL.APAR.UA37111.TXT                           4514/07  4514/07
________    "                                           4514/02  4514/06
________ CBL.APAR.UK27934.PTF.PACKED                    9722/00  9889/14
________ CBL.BA.CBLI.SYSTEM.INI                          013/00   013/00
________ CBL.BA.EXE                                     5065/04  5065/08
________    "                                            051/06   051/10
________    "                                           5092/00  5092/04
________    "                                           5081/09  5081/13
________    "                                           5065/09  5065/13
________    "                                           5093/00  5093/04
________    "                                           5092/10  5092/14
________    "                                           5092/05  5092/09
________    "                                            050/10   051/01
Line 1 of 813 | Col 1 of 199 | Views 1 | select * sort DSN
```

Figure 81. CBLVCAT LISTVTOC Raw Data Window.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command "M" if entry is a PDS/PDSE library, prefix line command "E" otherwise.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry. (Default for non-PDS/PDSE entries)
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as FSU - File Search/Update Window output.
FS	Open the File Search window for the entry.
I	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default for PDS/PDSE entries)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.

T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

For LISTVTOC Output, the data displayed is:

Name	Type	Description
DSN	Char	Data Set Name
CYL/HD	Char	Low and high Cylinder/Head Limits
CISIZE	Char	Control Interval Size
START	Char	Relative track/block start address
ALLOC	Char	Number of allocated tracks/blocks
USED	Char	Number of used tracks/blocks
TYPE	Char	Data set type
EXPIRES	Char	Expiry date
BLKSIZE	Char	Blocksize
LRECL	Char	Logical Record Length
RECFM	Char	Record Format
CREATED	Char	Creation date
INFO	Char	Informational messages
VOLUME	Char	VTOC volume id
ACCESSED	Char	Last Accessed date
UNIT	Char	Unit (cuu) of DASD volume

For LISTVCAT Output, the data displayed is:

Name	Type	Description
DSN	Char	Data Set Name
TYPE	Char	Data set type
NRECS	Char	Number of records
PCNT	Char	Percent of allocated space used
ALLOCT	Char	Total allocated tracks/blocks
ALLOCU	Char	Unused allocated tracks/blocks
ALLOCP	Char	Defined Primary allocation (tracks/blocks)
ALLOCS	Char	Defined Secondary allocation (tracks/blocks)
FRSP	Char	Defined Free Space per CI and CA
LMAX	Char	Defined Maximum Record Length
KL/BLK/IMB	Char	Duplicate of fields KL,RKP **or** BLKSIZE **or** IMB/REP
CISIZE	Char	Control Interval Size
BUFSP/IXL	Char	Duplicate of fields BUFSP **or** IXL
EXCPS	Char	Number of EXecuted Channel Programs
TIMESTMP	Char	Timestamp that file was last closed
NSEC	Char	Number of secondary extents
AVRL	Char	Defined average RECORDSIZE
PHYREC	Char	Physical Record Size
RECSTATS	Char	Records deleted, inserted, updated and read
KL	Char	Defined KEYS Length
RKP	Char	Defined KEYS Position
BLKSIZE	Char	Block size (VSE/VSAM SAM)

IMB/REP	Char	IMBED and/or REPLICATE flags
BUFSP	Char	Defined BUFFERSPACE
IXL	Char	Number of Index Levels
CI/CA	Char	Number of Control Intervals per Control Area
SHR	Char	Defined SHAREOPTIONS (local and cross system)
S/C	Char	Defined SHAREOPTIONS (local only) and USECLASS (primary only)
DEFINED	Char	Date on which file was defined
EXPIRES	Char	Date on which file expires
SPLITCI	Char	Number of Control Interval splits
SPLITCA	Char	Number of Control Area splits
SEVL	Char	Highest CBLVCAT severity message level
VOLUME	Char	Catalog Volume
GMAX	Char	GDG Maximum Level
GVER	Char	GDG Version number
GGEN	Char	GDG Generation number
STD1	Char	Reserved (blank)
STD2	Char	Reserved (blank)
HIUSERBA	Char	High Used Relative Byte Address
HIALLRBA	Char	High Allocated Relative Byte Address
FREEBYTES	Char	Number of unused allocated bytes
COMPONENT	Char	DATA or INDEX component DSN
ENTRY	Char	VSAM CLUSTER entry DSN
SMSS	Char	Defined SMS Storage Class
SMSD	Char	Defined SMS Data Class
SMSM	Char	Defined SMS Management Class
EXT	Char	Extended Attributes
CATALOG	Char	Catalog DSN

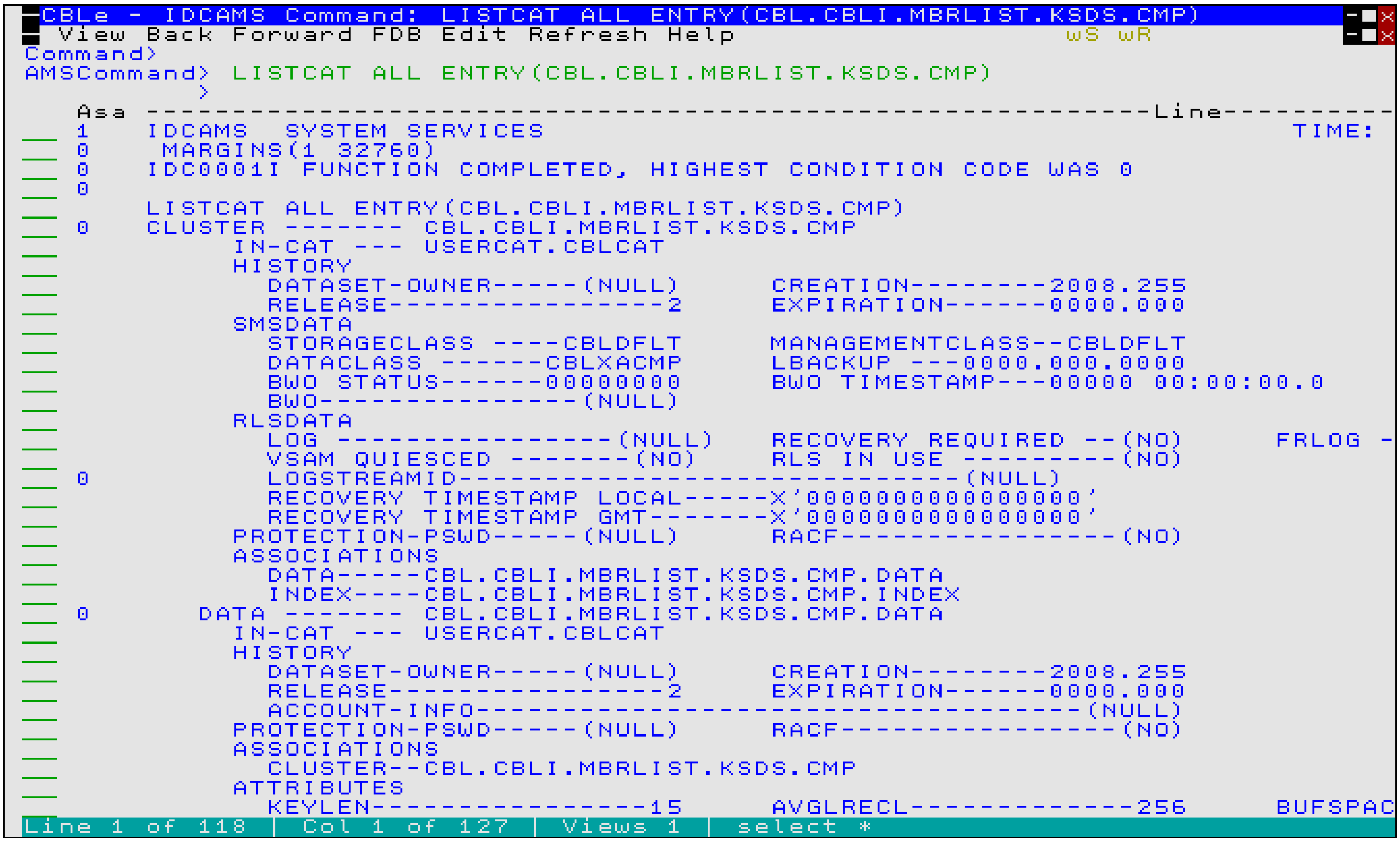
Execute IDCAMS (=8.3)

The IDCAMS Command window may be opened via the following:

- Select option 3. 'IDCAMS' from the Utilities Menu
- Select 'Execute IDCAMS' from the File menu in the CBLe main window menu bar.
- Enter the command AMS on the command line of any window.
- Enter the prefix command "I" where supported by a List type window.

The IDCAMS Command window allows the user to enter any IDCAMS command and view the output in the the window display area.

The IDCAMS Command window is essentially a List window and has the same charactersitics as List windows. For example filtering is supported to display new views of the data.



```
 CBLe - IDCAMS Command: LISTCAT ALL ENTRY(CBL.CBLI.MBRLIST.KSDS.CMP)
  View Back Forward FDB Edit Refresh Help                              wS wR
Command>
AMSCommand> LISTCAT ALL ENTRY(CBL.CBLI.MBRLIST.KSDS.CMP)
          >
   Asa ----------------------------------------------------------Line-----------
___ 1   IDCAMS  SYSTEM SERVICES                                                TIME:
___ 0    MARGINS(1 32760)
___ 0   IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
___ 0
___     LISTCAT ALL ENTRY(CBL.CBLI.MBRLIST.KSDS.CMP)
___ 0   CLUSTER ------- CBL.CBLI.MBRLIST.KSDS.CMP
___          IN-CAT --- USERCAT.CBLCAT
___          HISTORY
___            DATASET-OWNER-----(NULL)     CREATION--------2008.255
___            RELEASE----------------2     EXPIRATION------0000.000
___          SMSDATA
___            STORAGECLASS ----CBLDFLT     MANAGEMENTCLASS--CBLDFLT
___            DATACLASS -------CBLXACMP    LBACKUP ---0000.000.0000
___            BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
___            BWO---------------(NULL)
___          RLSDATA
___            LOG ----------------(NULL)   RECOVERY REQUIRED --(NO)     FRLOG -
___            VSAM QUIESCED -------(NO)    RLS IN USE ---------(NO)
___ 0          LOGSTREAMID-----------------------------(NULL)
___            RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
___            RECOVERY TIMESTAMP GMT-------X'0000000000000000'
___          PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
___          ASSOCIATIONS
___            DATA-----CBL.CBLI.MBRLIST.KSDS.CMP.DATA
___            INDEX----CBL.CBLI.MBRLIST.KSDS.CMP.INDEX
___ 0      DATA ------- CBL.CBLI.MBRLIST.KSDS.CMP.DATA
___          IN-CAT --- USERCAT.CBLCAT
___          HISTORY
___            DATASET-OWNER-----(NULL)     CREATION--------2008.255
___            RELEASE----------------2     EXPIRATION------0000.000
___            ACCOUNT-INFO-----------------------------------(NULL)
___          PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
___          ASSOCIATIONS
___            CLUSTER--CBL.CBLI.MBRLIST.KSDS.CMP
___          ATTRIBUTES
               KEYLEN-----------------15     AVGLRECL-------------256     BUFSPAC
Line 1 of 118 | Col 1 of 127 | Views 1 | select *
```

Figure 82. IDCAMS Command window.

Panel Fields

AMSCommand>
Specify valid IDCAMS command syntax.

Prefix Commands

No prefix line commands are supported for IDCAMS Command windows.

Columns Displayed

Asa	Char	ASA print control character
Line	Char	Print line

Execute POWER

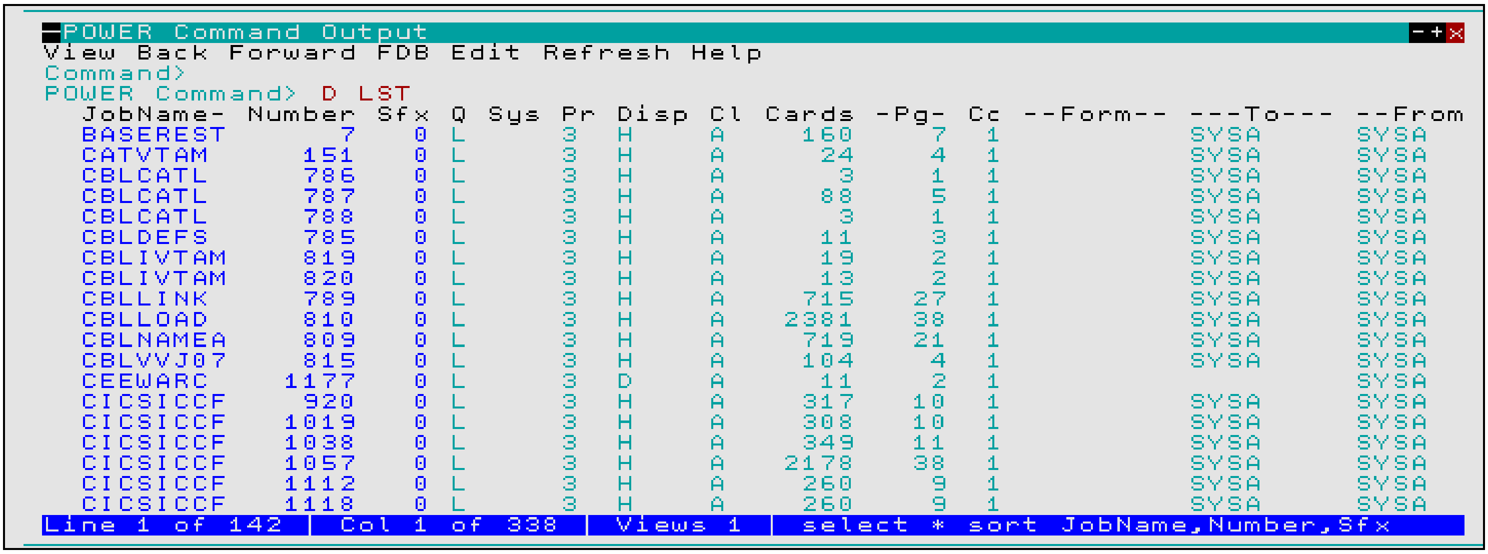
The POWER Command Output window may be opened via the following:

- Select 'Execute POWER' from the File menu in the CBLe main window menu bar.
- Enter command POWER on the command line of any window.

The POWER Command Output window allows the user to enter VSE POWER commands and view the output in the window display area.

If SELCOPY/i INI variables System.VSESMLogon=No (i.e. no Security Manager is active) and System.TrustedUser=No, then POWER commands are restricted to PDISPLAY operations only.

The POWER Command Output window is essentially a List window and has the same characteristics as List windows. For example select, sort and filter to display new views of the data are supported.



```
 POWER Command Output                                                      -+x
View Back Forward FDB Edit Refresh Help
Command>
POWER Command> D LST
  JobName- Number Sfx Q Sys Pr Disp Cl Cards -Pg- Cc --Form-- ---To--- --From
  BASEREST      7   0 L     3  H    A    160    7   1          SYSA     SYSA
  CATVTAM     151   0 L     3  H    A     24    4   1          SYSA     SYSA
  CBLCATL     786   0 L     3  H    A      3    1   1          SYSA     SYSA
  CBLCATL     787   0 L     3  H    A     88    5   1          SYSA     SYSA
  CBLCATL     788   0 L     3  H    A      3    1   1          SYSA     SYSA
  CBLDEFS     785   0 L     3  H    A     11    3   1          SYSA     SYSA
  CBLIVTAM    819   0 L     3  H    A     19    2   1          SYSA     SYSA
  CBLIVTAM    820   0 L     3  H    A     13    2   1          SYSA     SYSA
  CBLLINK     789   0 L     3  H    A    715   27   1          SYSA     SYSA
  CBLLOAD     810   0 L     3  H    A   2381   38   1          SYSA     SYSA
  CBLNAMEA    809   0 L     3  H    A    719   21   1          SYSA     SYSA
  CBLVVJ07    815   0 L     3  H    A    104    4   1          SYSA     SYSA
  CEEWARC    1177   0 L     3  D    A     11    2   1                   SYSA
  CICSICCF    920   0 L     3  H    A    317   10   1          SYSA     SYSA
  CICSICCF   1019   0 L     3  H    A    308   10   1          SYSA     SYSA
  CICSICCF   1038   0 L     3  H    A    349   11   1          SYSA     SYSA
  CICSICCF   1057   0 L     3  H    A   2178   38   1          SYSA     SYSA
  CICSICCF   1112   0 L     3  H    A    260    9   1          SYSA     SYSA
  CICSICCF   1118   0 L     3  H    A    260    9   1          SYSA     SYSA
Line 1 of 142 |  Col 1 of 338 |  Views 1 |  select * sort JobName,Number,Sfx
```

Figure 83. POWER Command Output window for PDISPLAY LST.

Panel Fields

POWER Command>
Specify the VSE POWER command. Note that POWER commands relating to cross partition usage (e.g. PDISPLAY STATUS) are not supported.

Prefix Commands

The following prefix line commands are supported for **PDISPLAY** (RDR, LST or PUN) output only:

Command	Description
<Dflt>	Prefix Line command E.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry. If an entry is password protected, then it may be edited by any user so long as the password is supplied. A pop-up window will prompt the user for the password. Non-password protected entries may only be edited if either of the following are true: • System.TrustedUser=Yes in the SELCOPY/i INI file. • System.VSESMLogon=Yes in the SELCOPY/i INI file and the TO or FROM attributes match the current user's userid. Note that VSE Basic Security Manager (BSM) alone does not impose access restrictions on the VSE POWER queues.
K	Delete (Kill) the entry without prompting for verification.

Columns Displayed

The data displayed for PDISPLAY ALL/LST/PUN/RDR is:

Name	Type	Description
JobName	Char	JOB NAME
Number	UInt	JOB NUMBER
Sfx	UInt	JOB SUFFIX NUMBER
Q	Char	QUEUE IDENTIFIER (R, L, P)
Sys	Char	SYSTEM ID. (TARGET/PROCESS.)
Pr	Char	PRIORITY
Disp	Char	DISPOSITION (*..IN EXEC.)
Cl	Char	CLASS
Cards	UInt	NUMBER OF RECORDS SPOOLED
Pg	UInt	NUMBER OF PAGES SPOOLED
Cc	UInt	NUMBER OF COPIES
Form	Char	FORMS IDENTIFIER
To	Char	TARGET DESTINATION USER/REMOTE ID
From	Char	ORIGINATING USER/REMOTE ID
Cn	Char	CENTURY OF CREATION DATE
Date	Char	CREATION DATE OF QUEUE ENTRY
Start	Dec	START TIME (0HHMMSSF)
Stop	Dec	STOP TIME (0HHMMSSF)
PXFMRLEN	UInt	RECORD LENGTH
PXFMTYPE	Hex	RECORD TYPE
PXFMVOL	Hex	TAPE BAM VOLUME NUMBER
PXFMUSER	Char	USER INFORMATION
PXFMFLG1	Hex	CONTROL FLAG 1
PXFMRCFM	Hex	RECORD FORMAT
PXFMSTAT	Char	PAPER STATUS BYTE
PXFMLNE#	UInt	NUMBER OF LINES/CARDS SPOOLED
PXFMFLSH	Char	FLASH IDENTIFIER
PXFMCPYG	Hex	COPY GROUPINGS
PXFMFLG2	Hex	CONTROL FLAG 2
PXFMNSEP	UInt	NUMBER OF SEP. PAGES / CARDS
PXFMJBO#	UInt	ORIGINAL JOB NUMBER
PXFMCMPT	Char	COMPACTION TABLE NAME
PXFMNODE	Char	TARGET DESTINATION NODE NAME
PXFMORGN	Char	ORIGINATING NODE NAME
PXFMSUBS	Char	SUBSYSTEM NAME (EXTERNAL WRITER ID)
PXFMDDND	Char	NEXT DUE DATE
PXFMDDNT	Char	NEXT DUE TIME
PXFMQNUM	UInt	QUEUE ENTRY NUMBER
PXFMSECN	Char	QUEUE ENTRY SECURITY ZONE (SECNODE)
PXFMDIST	Char	OUTPUT DISTRIBUTION CODE
PXFMMACN	UInt	.. NON SHARED ACCESS COUNT
PXFMMAC1	UInt	.. SHARED SYSID 1 ACC. CNT.
PXFMMAC2	UInt	.. SHARED SYSID 2 ACC. CNT.
PXFMMAC3	UInt	.. SHARED SYSID 3 ACC. CNT.

PXFMMAC4	UInt	.. SHARED SYSID 4 ACC. CNT.
PXFMMAC5	UInt	.. SHARED SYSID 5 ACC. CNT.
PXFMMAC6	UInt	.. SHARED SYSID 6 ACC. CNT.
PXFMMAC7	UInt	.. SHARED SYSID 7 ACC. CNT.
PXFMMAC8	UInt	.. SHARED SYSID 8 ACC. CNT.
PXFMMAC9	UInt	.. SHARED SYSID 9 ACC. CNT.

The data displayed for other POWER commands is:

Name	Type	Description
Text	Char	Power Display Output

Define Catalog ALIAS (=8.4)

The Define Catalog ALIAS Dialog window may be opened via the following:

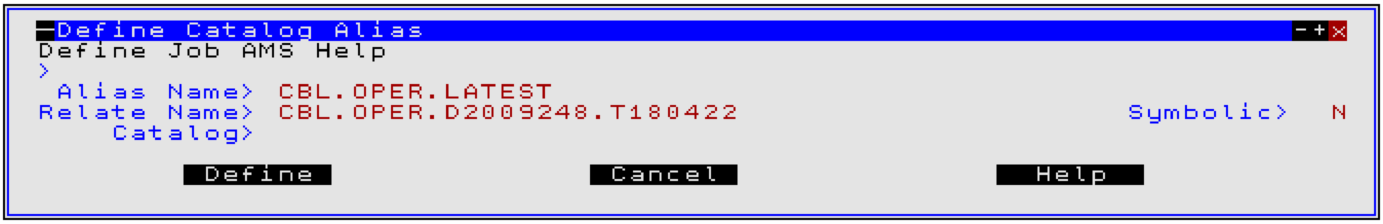
- Select option 4. 'Catalog ALIAS' from the Utilities Menu or select option 6. 'Alias' from the Create New Datasets menu panel.
- Select 'Create Catalog ALIAS' from the File menu in the CBLe main window menu bar.
- Enter command AMSA on the command line of any window.
- Enter the List window prefix command "A" against a non-VSAM data set list entry.

The Define Catalog ALIAS dialog provides a simple interface for the user to supply IDCAMS DEFINE ALIAS characteristics for a new alias name. Aliases may be defined for non-VSAM data sets and, if the user has sufficient authority, user catalogs. Compare with the Create Library ALIAS dialog window which creates PDS/PDSE library member aliases.

Select the appropriate menu bar item (see below) to define the new entry.

Fields within this dialog represents the relevant IDCAMS DEFINE ALIAS parameters as appropriate for the entry being defined. Please refer to *"DFSMS Access Method Services for Catalogs"* for further information.

Note: Not implemented for CMS and VSE.



```
 Alias Name> CBL.OPER.LATEST
Relate Name> CBL.OPER.D2009248.T180422                      Symbolic>   N
    Catalog>
```

Figure 84. Define Catalog ALIAS Dialog window.

Menu Bar Items

`Define`
: Start the VSAM object definition. (Foreground)

`Job`
: Creates and edits the IDCAMS DEFINE statement including job control ready for submission to batch. (See CBLe command SUBMIT.) (Background)

AMS
: Opens a CBLe edit view containing generated AMS command syntax to perform the IDCAMS DEFINE. Execute by placing the cursor on the first line of the command and hitting <PF4> The command may be copied to the user's HOME command centre for future reference.

Help
: Open the help window for the Define Catalog ALIAS dialog window.

Panel Fields

`Alias Name>`
: Name of the ALIAS object to be defined.

Relate Name>
Name of the object to which the ALIAS will relate.

Catalog>
This field entry specifies the catalog in which the alias is to be defined. If the alias is for a user catalog connector, this field should contain the name of the master catalog. Please refer to section *"Catalog Selection Order for DEFINE"* in *"DFSMS Access Method Services for Catalogs"* for catalog selection when this field is null.

Symbolic>
This field entry may be "Y" or "N" to indicate that the **Relate>** field is a SYMBOLICRELATE containing system symbols (i.e. an Extended ALIAS is to be defined.) See *"DFSMS Managing Catalogs"* for further information.

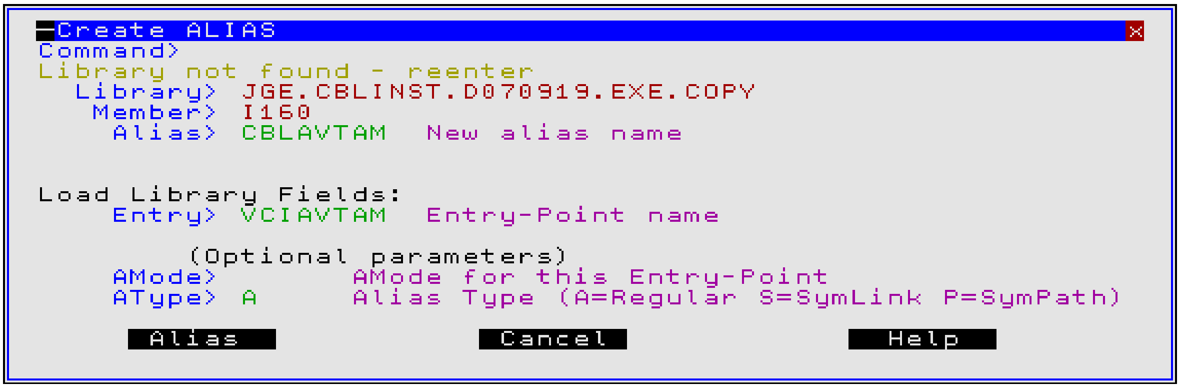
Create Library ALIAS (=8.5)

The Create Library ALIAS Dialog window may be opened via the following:

- Select option 5. 'Library ALIAS' from the Utilities Menu.
- Select 'Create Library ALIAS' from the File menu in the CBLe main window menu bar.
- Enter command ALIAS -DLG on the command line of any window.
- Enter the List window prefix command "A" against an entry in a Library List.

The Create Library ALIAS dialog provides a simple interface to create a new PDS or PDSE library member alias. Compare with the Define Catalog ALIAS dialog window which creates cataloged aliases for non-VSAM data sets and user catalogs.

Note that aliases for PDSE load-library members are created using the binder to relink the module being aliased. This will result in an update to the module's **TTR**.



```
Create ALIAS
Command>
Library not found - reenter
   Library> JGE.CBLINST.D070919.EXE.COPY
     Member> I160
       Alias> CBLAVTAM   New alias name


Load Library Fields:
       Entry> VCIAVTAM   Entry-Point name

             (Optional parameters)
       AMode>            AMode for this Entry-Point
       AType> A          Alias Type (A=Regular S=SymLink P=SymPath)
```

Figure 85. Create Library ALIAS Dialog window.

Panel Fields

Library>
The DSN of the PDS(E) library. (This may be a LOAD Library.)

Member>
The library member name for which an alias will be generated.

Alias>
The new alias name to be generated.

Entry>
For load library aliases only, the symbolic name of the entry-point address to be used.

AMode>
For a load library aliases only, the Addressing Mode for the entry point specified in Entry>. Valid arguments are 24, 31 and ANY.

AType>
For a load library aliasses only, the alias type to be generated. Valid arguments are A=Regular, S=SymLink, P=SymPath.

Execute IEBCOPY (=8.6)

The IEBCOPY Dialog window may be opened via the following:

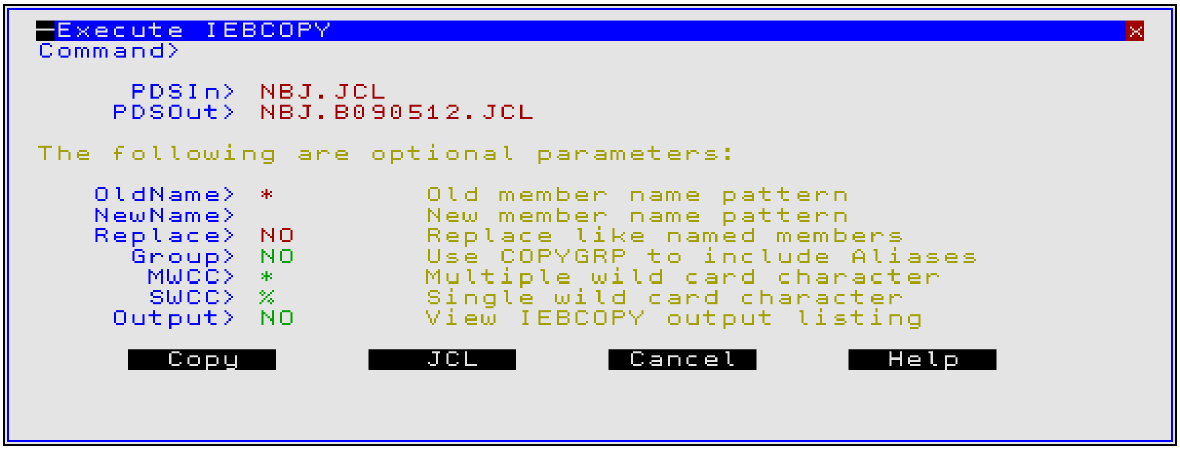
- Select option 6. 'IEBCOPY' from the Utilities Menu.
- Select 'Execute IEBCOPY' from the File menu in the CBLe main window menu bar.
- Enter command IEBCOPYDIALOG on the command line of any window.
- Enter the List window prefix command "C" against a PDS(E) entry in a Dataset List or Catalog List, or any entry within in a Library List.

The IEBCOPY Dialog provides an intuitive interface to copy PDS(E) libraries or individual members to a new or existing target library.

Select "Copy" to perform the IEBCOPY in the foreground or "JCL" to generate a batch job stream in a CBLe text edit view. Having selected "JCL" the user can issue "JOBCARD" to insert a skeleton JOB statement, before executing SUB to submit the job to batch.

Note: Unless already positioned on one of the window buttons (Copy, JCL, Cancel or Help), <Enter> will first position the cursor on the "Copy" button, <Enter> a second time will select (press) the button to action the command.

Any non-zero return code encountered using the foreground "Copy" option will open the Execute IEBCOPY output listing displaying the SYSPRINT output.
Unless the Output> field value is "YES", if a zero return code is encountered, no output window is opened and a message reporting the number of members copied is returned.



```
Execute IEBCOPY
Command>

     PDSIn>  NBJ.JCL
    PDSOut>  NBJ.B090512.JCL

The following are optional parameters:

   OldName>  *          Old member name pattern
   NewName>             New member name pattern
   Replace>  NO         Replace like named members
     Group>  NO         Use COPYGRP to include Aliases
      MWCC>  *          Multiple wild card character
      SWCC>  %          Single wild card character
    Output>  NO         View IEBCOPY output listing
```

Figure 86. IEBCOPY Dialog window.

Panel Fields

PDSIn>

Specify the DSN of the source PDS(E) library. (This may be a LOAD Library.)

PDSOut>

Specify the DSN of the target PDS(E) library. This may be the same library DSN specified for PDSIn.

Default is the value specified on the last invocation of the IEBCOPY dialog. Otherwise, the default is the PDSIn value.

OldName>

Source library member name mask identifying members to be copied.

Multiple character and single character wild cards, defined by the MWCC and SWCC values (see below) may be used in the member mask.

If no value is specified (i.e. unset), the entire library will be copied. This is different to simply specifying wildcard "*" (asterisk) which copies all members individually. To copy all members of a library, the process is quicker if the value is blank.

If invoked via the "C" prefix command, default is the value of the Library List "Entry" field or is unset if a Dataset or Catalog List. Otherwise, the default is the value specified on the last invocation of the IEBCOPY dialog.

NewName>

Target member name. The member name specified in OldName will be renamed to this new name.

This field must be empty if a wildcard character is used in the OldName member mask. Wild cards are not supported for NewName.
Default is the OldName value.

Replace>

Enter "YES" or "NO" to indicate whether existing members in the target library are to be replaced if the source and target member names match.
Default is the value specified on the last invocation of the IEBCOPY dialog, otherwise "NO".

Group>

Enter "YES" or "NO" to indicate whether any defined ALIAS entries for the selected library members are to be copied also. Note that this also applies to Load Library members.

MWCC>

Specifies the Multiple Wild Card Character which represents zero or more characters in the OldName library member mask.
Default is "*" (asterisk).

SWCC>

Specifies the Single Wild Card Character which represents one character in the OldName library member mask.
Default is "%" (percent).

Output>

Enter "YES" or "NO" to indicate whether the IEBCOPY SYSPRINT output is to be displayed.
Default is "NO".

DB2 Dynamic SQL

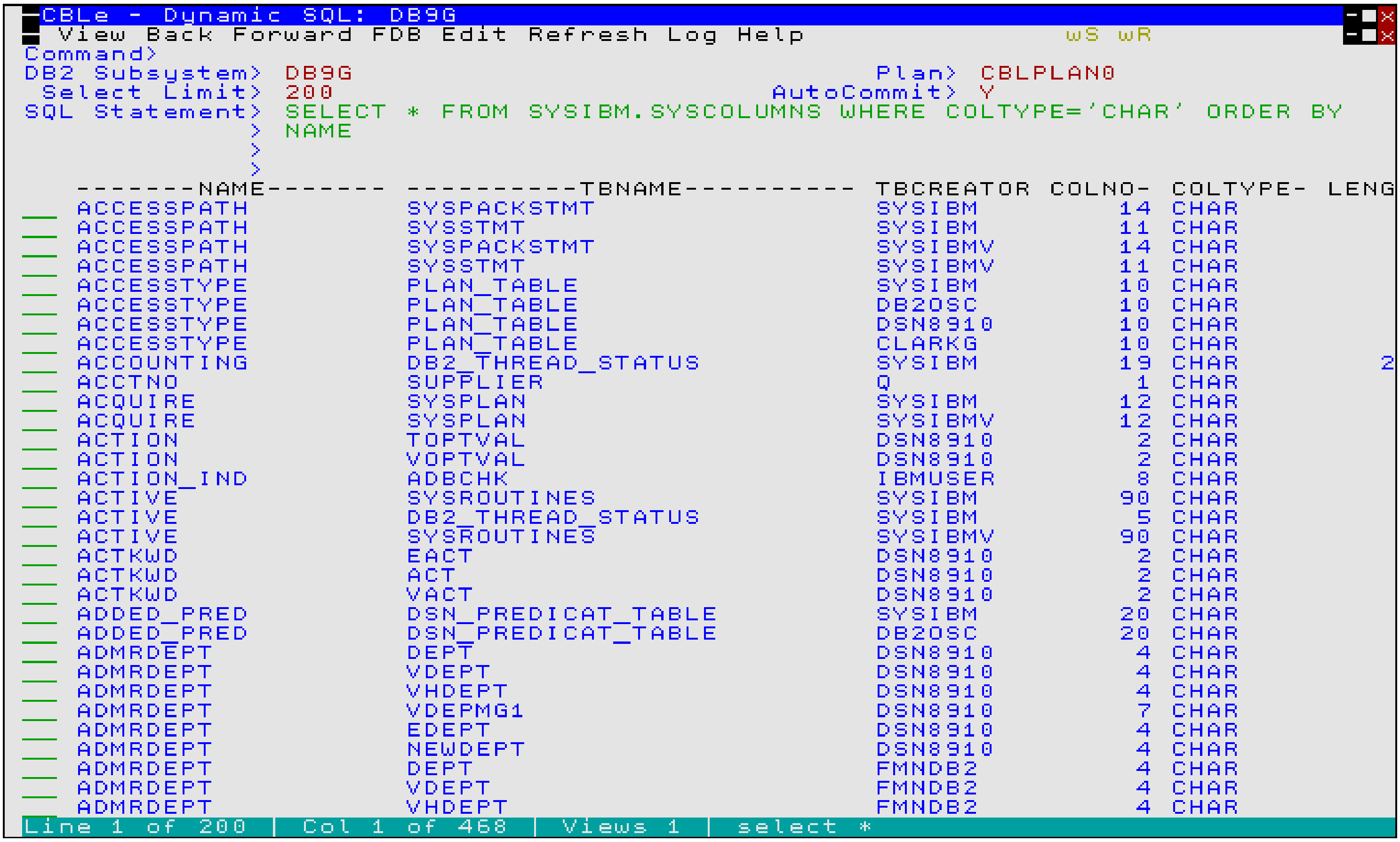
The DB2 Dynamic SQL window is a list window, opened on execution of the SQL command.

DB2 Dynamic SQL window may be used to submit SQL commands to a DB2 subsystem and display the resultant messages and/or table views.

Note that SQL statements may also be executed via the Execute SQL Statements panel as part of the SELCOPY/i DB2 interface.

The DB2 Dynamic SQL capability is available only to MVS sites where **SELCOPY** is installed. The SELCOPY DB2 interface, **SELCOPQL** load module is used to pass SQL statements to the DB2 data base and so must also be available in the SELCOPY load library.

On startup, the Dynamic SQL window connects the user to the DB2 subsystem using the DB2 subsystem name and plan specified in the **CBLNAME** load module.

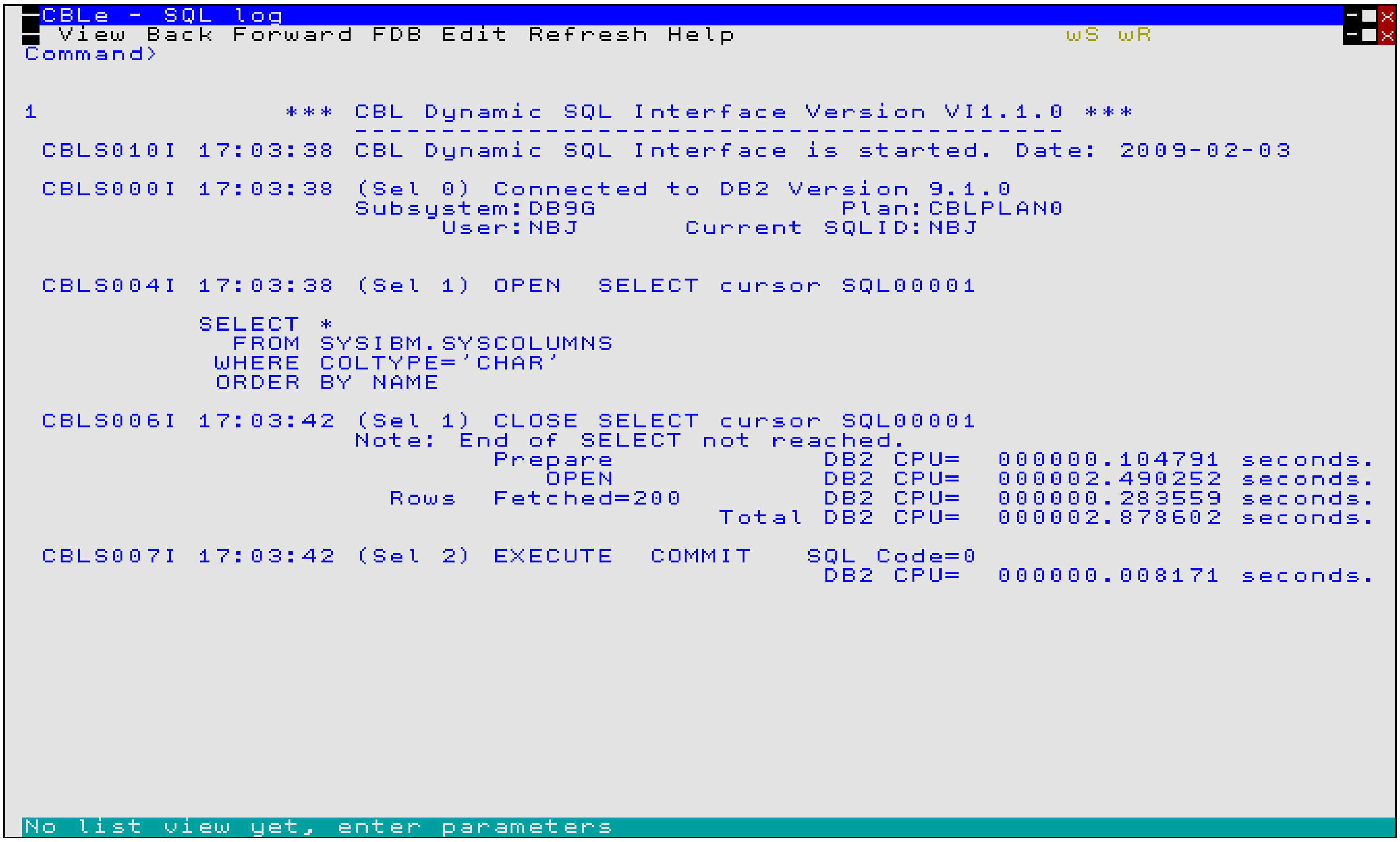


```
 CBLe - Dynamic SQL: DB9G
  View Back Forward FDB Edit Refresh Log Help                 wS wR
Command>
DB2 Subsystem> DB9G                                  Plan> CBLPLAN0
 Select Limit> 200                             AutoCommit> Y
SQL Statement> SELECT * FROM SYSIBM.SYSCOLUMNS WHERE COLTYPE='CHAR' ORDER BY
             > NAME
             >
             >
   -------NAME------- ----------TBNAME---------- TBCREATOR COLNO- COLTYPE- LENG
__ ACCESSPATH         SYSPACKSTMT                SYSIBM        14 CHAR
__ ACCESSPATH         SYSSTMT                    SYSIBM        11 CHAR
__ ACCESSPATH         SYSPACKSTMT                SYSIBMV       14 CHAR
__ ACCESSPATH         SYSSTMT                    SYSIBMV       11 CHAR
__ ACCESSTYPE         PLAN_TABLE                 SYSIBM        10 CHAR
__ ACCESSTYPE         PLAN_TABLE                 DB2OSC        10 CHAR
__ ACCESSTYPE         PLAN_TABLE                 DSN8910       10 CHAR
__ ACCESSTYPE         PLAN_TABLE                 CLARKG        10 CHAR
__ ACCOUNTING         DB2_THREAD_STATUS          SYSIBM        19 CHAR        2
__ ACCTNO             SUPPLIER                   Q              1 CHAR
__ ACQUIRE            SYSPLAN                    SYSIBM        12 CHAR
__ ACQUIRE            SYSPLAN                    SYSIBMV       12 CHAR
__ ACTION             TOPTVAL                    DSN8910        2 CHAR
__ ACTION             VOPTVAL                    DSN8910        2 CHAR
__ ACTION_IND         ADBCHK                     IBMUSER        8 CHAR
__ ACTIVE             SYSROUTINES                SYSIBM        90 CHAR
__ ACTIVE             DB2_THREAD_STATUS          SYSIBM         5 CHAR
__ ACTIVE             SYSROUTINES                SYSIBMV       90 CHAR
__ ACTKWD             EACT                       DSN8910        2 CHAR
__ ACTKWD             ACT                        DSN8910        2 CHAR
__ ACTKWD             VACT                       DSN8910        2 CHAR
__ ADDED_PRED         DSN_PREDICAT_TABLE         SYSIBM        20 CHAR
__ ADDED_PRED         DSN_PREDICAT_TABLE         DB2OSC        20 CHAR
__ ADMRDEPT           DEPT                       DSN8910        4 CHAR
__ ADMRDEPT           VDEPT                      DSN8910        4 CHAR
__ ADMRDEPT           VHDEPT                     DSN8910        4 CHAR
__ ADMRDEPT           VDEPMG1                    DSN8910        7 CHAR
__ ADMRDEPT           EDEPT                      DSN8910        4 CHAR
__ ADMRDEPT           NEWDEPT                    DSN8910        4 CHAR
__ ADMRDEPT           DEPT                       FMNDB2         4 CHAR
__ ADMRDEPT           VDEPT                      FMNDB2         4 CHAR
   ADMRDEPT           VHDEPT                     FMNDB2         4 CHAR
Line 1 of 200 | Col 1 of 468 | Views 1 | select *
```

Figure 87. DB2 Dynamic SQL window.

The contents of the display area may be edited in a new file using the CBLe text editor by selecting **Edit** from the window menu.

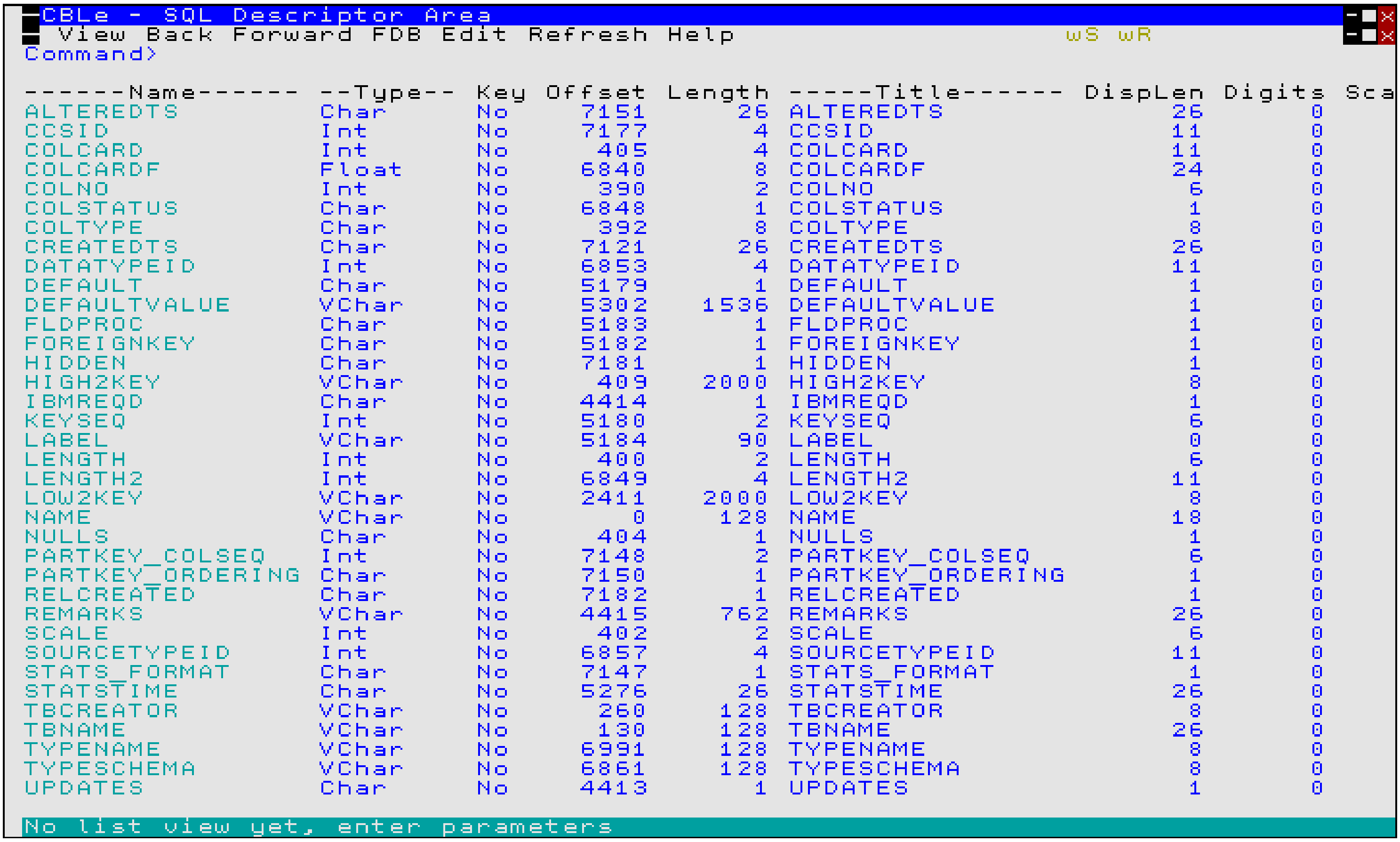
Select **Log** from the window menu to open the SQL Log output which displays the daignosis information and SQL return codes passed from the DB2 sub-system.



```
1              *** CBL Dynamic SQL Interface Version VI1.1.0 ***
                   -----------------------------------------
 CBLS010I 17:03:38 CBL Dynamic SQL Interface is started. Date: 2009-02-03

 CBLS000I 17:03:38 (Sel 0) Connected to DB2 Version 9.1.0
                   Subsystem:DB9G              Plan:CBLPLAN0
                        User:NBJ      Current SQLID:NBJ


 CBLS004I 17:03:38 (Sel 1) OPEN  SELECT cursor SQL00001

          SELECT *
            FROM SYSIBM.SYSCOLUMNS
           WHERE COLTYPE='CHAR'
           ORDER BY NAME

 CBLS006I 17:03:42 (Sel 1) CLOSE SELECT cursor SQL00001
                   Note: End of SELECT not reached.
                           Prepare            DB2 CPU=  000000.104791 seconds.
                              OPEN            DB2 CPU=  000002.490252 seconds.
                     Rows  Fetched=200        DB2 CPU=  000000.283559 seconds.
                                        Total DB2 CPU=  000002.878602 seconds.

 CBLS007I 17:03:42 (Sel 2) EXECUTE  COMMIT   SQL Code=0
                                              DB2 CPU=  000000.008171 seconds.









No list view yet, enter parameters
```

Figure 88. DB2 Dynamic SQL LOG window.

Select **FDB** from the window menu to open the Field Descriptor Block which provides detailed information on each field displayed by an SQL SELECT statement.



```
CBLe - SQL Descriptor Area
 View Back Forward FDB Edit Refresh Help                          wS wR
Command>

------Name------ --Type-- Key Offset Length -----Title------ DispLen Digits Sca
ALTEREDTS        Char     No    7151     26 ALTEREDTS             26      0
CCSID            Int      No    7177      4 CCSID                 11      0
COLCARD          Int      No     405      4 COLCARD               11      0
COLCARDF         Float    No    6840      8 COLCARDF              24      0
COLNO            Int      No     390      2 COLNO                  6      0
COLSTATUS        Char     No    6848      1 COLSTATUS              1      0
COLTYPE          Char     No     392      8 COLTYPE                8      0
CREATEDTS        Char     No    7121     26 CREATEDTS             26      0
DATATYPEID       Int      No    6853      4 DATATYPEID            11      0
DEFAULT          Char     No    5179      1 DEFAULT                1      0
DEFAULTVALUE     VChar    No    5302   1536 DEFAULTVALUE           1      0
FLDPROC          Char     No    5183      1 FLDPROC                1      0
FOREIGNKEY       Char     No    5182      1 FOREIGNKEY             1      0
HIDDEN           Char     No    7181      1 HIDDEN                 1      0
HIGH2KEY         VChar    No     409   2000 HIGH2KEY               8      0
IBMREQD          Char     No    4414      1 IBMREQD                1      0
KEYSEQ           Int      No    5180      2 KEYSEQ                 6      0
LABEL            VChar    No    5184     90 LABEL                  0      0
LENGTH           Int      No     400      2 LENGTH                 6      0
LENGTH2          Int      No    6849      4 LENGTH2               11      0
LOW2KEY          VChar    No    2411   2000 LOW2KEY                8      0
NAME             VChar    No       0    128 NAME                  18      0
NULLS            Char     No     404      1 NULLS                  1      0
PARTKEY_COLSEQ   Int      No    7148      2 PARTKEY_COLSEQ         6      0
PARTKEY_ORDERING Char     No    7150      1 PARTKEY_ORDERING       1      0
RELCREATED       Char     No    7182      1 RELCREATED             1      0
REMARKS          VChar    No    4415    762 REMARKS               26      0
SCALE            Int      No     402      2 SCALE                  6      0
SOURCETYPEID     Int      No    6857      4 SOURCETYPEID          11      0
STATS_FORMAT     Char     No    7147      1 STATS_FORMAT           1      0
STATSTIME        Char     No    5276     26 STATSTIME             26      0
TBCREATOR        VChar    No     260    128 TBCREATOR              8      0
TBNAME           VChar    No     130    128 TBNAME                26      0
TYPENAME         VChar    No    6991    128 TYPENAME               8      0
TYPESCHEMA       VChar    No    6861    128 TYPESCHEMA             8      0
UPDATES          Char     No    4413      1 UPDATES                1      0

No list view yet, enter parameters
```

Figure 89. DB2 SQL FDB Window.

Panel Fields

`DB2 Subsystem>`

Specify the DB2 sub-system name to be the target of the CONNECT.
Changing the Subsystem name will cause CBLi to disconnect the user from the current subsystem and reconnect to the new subsystem.

Default is that defined by the CBLiINI option, DB2.SSN, otherwise the sub-system name specified in the DB2SubSys field of the CBLNAME load module is used.

`Plan>`

Specify the SELCOPY (SELCOPQL) DB2 plan name which has been bound to the DB2 sub-system.
This is the name assigned to the SELCOPY plan during the BIND to the DB2 subsystem.

Default is that defined by the CBLiINI option, DB2.Plan, otherwise the plan name specified in the DB2Plan field of the CBLNAME load module is used.

`Select Limit>`

Limit the number of rows to be displayed in the Dynamic SQL window following a SELECT transaction. Once the limit threshold has been reached, a pop-up message window is displayed and no further attempt is made to retrieve selected rows of data.
The *n_rows* value is placed in the "Select Limit>" field of the Dynamic SQL window.
The default limit is that defined by the CBLiINI option, DB2.SelectLimit, otherwise no limit is implied.

`AutoCommit>`

Determine whether a COMMIT is to be automatically issued following every transaction (AutoCommit). If COMMIT=NO, then the user should issue COMMIT manually to commit any changes made to the data. A commit is executed automatically when the Dynamic SQL window is closed, regardless of the AutoCommit field setting.
The commit value is reflected in the "AutoCommit>" field of the Dynamic SQL window.
The default is YES.

`SQL Statement>`

Specify valid SQL statement syntax to be executed either directly or via an input control file.

If an input control file is specified, then the input fileid must be prefixed by "<" (less than). e.g. `< NBJ.SQL.CTL(SYSTABS)`

The resulting data or messages are displayed in the window display area.

Favourite Datasets/Commands (=8.7)

Overview

The Favourite Datasets/Commands utility (FAV) enables users to specify a default project hierarchy and also assign file names and command streams to items of a numbered list. This utility was introduced in order to assist migratation from other productivity software that offer similar features.

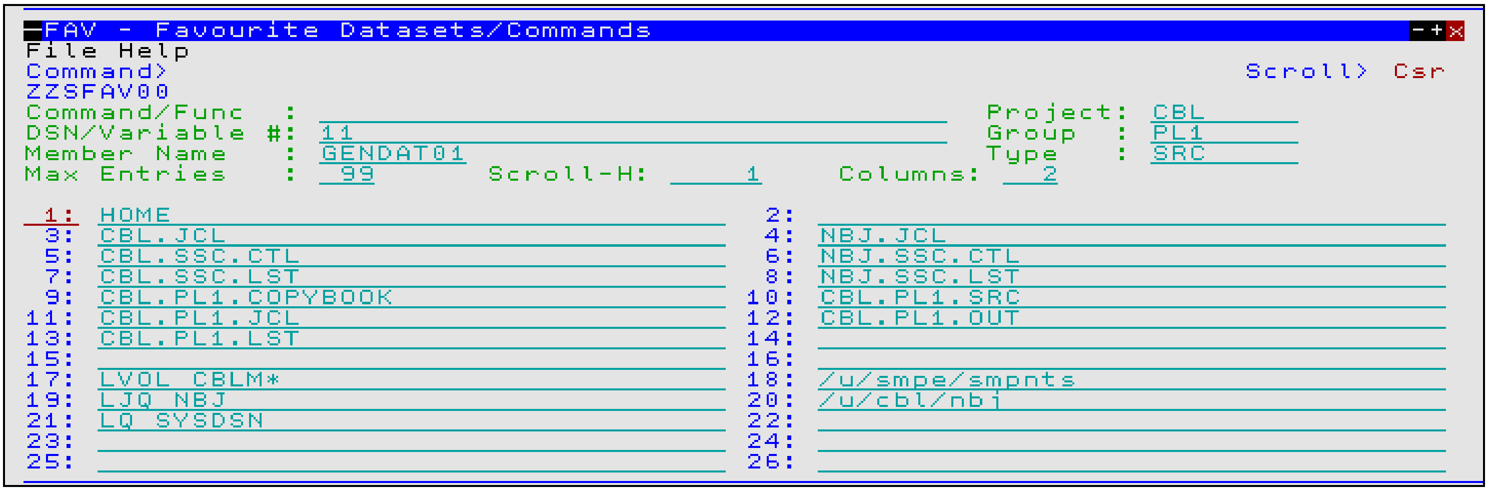
Users may configure numbered items and later reference the file name or execute the command assigned to an item, simply by entering the item number. This offers users an interface to commonly accessed data sets which be used in addition to commands entered in the user's HOME command centre (CMX) file.

Favourite Datasets/Commands Panel

The Favourite Datasets/Commands panel window (ZZSFAV00) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select option 7. 'Favourites' from the Utilities Menu.
- Select 'Favourites' from the Utilities menu in the CBLe main window menu bar.
- Enter command FAV on the command line of any window.

By default, field entries are populated with arguments and options that were entered the last time the Search for Favourite Datasets/Commands panel was used.



```
FAV - Favourite Datasets/Commands
File Help
Command>                                                       Scroll> Csr
ZZSFAV00
Command/Func  :                                          Project: CBL
DSN/Variable #: 11                                       Group  : PL1
Member Name   : GENDAT01                                 Type   : SRC
Max Entries   :  99        Scroll-H:      1     Columns:   2

 1: HOME                                  2:
 3: CBL.JCL                               4: NBJ.JCL
 5: CBL.SSC.CTL                           6: NBJ.SSC.CTL
 7: CBL.SSC.LST                           8: NBJ.SSC.LST
 9: CBL.PL1.COPYBOOK                     10: CBL.PL1.SRC
11: CBL.PL1.JCL                          12: CBL.PL1.OUT
13: CBL.PL1.LST                          14:
15:                                      16:
17: LVOL CBLM*                           18: /u/smpe/smpnts
19: LJQ NBJ                              20: /u/cbl/nbj
21: LQ SYSDSN                            22:
23:                                      24:
25:                                      26:
```

Figure 90. FAV - Favourite Datasets/Commands panel.

When <Enter> is pressed, the a command verb and a fileid parameter is constructed from one or more of the `DSN/Variable #`, `Member Name`, `Project`, `Group` and `Type` fields.

The fielid is determined based primarily on the contents of the `DSN/Variable #` field as follows:

Field Content	Fileid Determination
null	Use the contents of the Project, Group, Type and Member Name fields.
non-numeric	Use the contents of the DSN/Variable # and Member Name field.
numeric	Use the contents of the specified number list item and the Member Name field.

Note that, pressing <Enter> or, if configured, double-clicking the left mouse button on one of the numbered items, is equivalent to entering the item number in the **`DSN/Variable #`** field then pressing <Enter>.

Panel Input Fields

`Command/Func:`

This input field (ZCOMMAND) allows the user to optionally enter a command to be executed using the contents of the **`DSN/Variable #`** field and, if a PDS/PDSE or GDG DSN, **`Member Name`** field as input to the command. If the contents of the **`DSN/Variable #`** field is numeric, referencing one of the numbered list items, then the contents of that list item is used as input to the command.

If, however, the contents of the list item starts with a recognised command verb, then the command entered in the **`Command/Func:`** field is ignored and the list item command stream is executed instead.

If left empty, then the command generated for this field on pressing <Enter> is determined as follows:

Command Executed	Condition
none	A command is already included as part of the specified list item number.
LA	The fileid is a single token (qualifier) containing no "." (dot/period) and no leading "/" (slash).
EDIT	The fileid has an MVS PDS/PDSE DSN with a member name or is an HFS file path.
LL	The fileid is an MVS PDS/PDSE DSN with no member name.
LD	The fileid is not an MVS PDS/PDSE DSN.

`DSN/Variable #:`

This input field (ZFAVNUM) optionally specifies a complete fileid, the DSN of a PDS/PDSE library or a list item number that references a fileid or command stream.

If left empty, then a fileid is generated from the **`Project`**, **`Group`**, **`Type`** and **`Member Name`** fields.

Entering an invalid list item number in this field will return the following errors:

```
ZZSP025E Parameter n is invalid in command SaoGet n.
ZZSE043E RC=20 from: SAOSTRING
```

`Member Name:`

This input field (ZMEMBER) specifies a library member name to be included as part of the fileid.

For MVS systems only, where this field is not empty, the use of its contents in the resultant fileid is based on whether a member name has already been specified via the other fields used to resolve the fileid. i.e. If no member name is already identified within the constructed fileid, then the contents of the Member Name field are enclosed in "( )" (parentheses) and appended to the fileid.

For VSE and CMS, this member name is used only if the **DSN/Variable #** field is null, in which case the fileid is built from the Project, Group, Type and Member Name fields.

`Project:`
`Group:`
`Type:`

Input fields Project, Group and Type (ZPROJECT, ZGROUP, ZTYPE respectively) identify the default fileid tokens (qualifiers) to be used if the **DSN/Variable #** field is null.

For MVS, the Project, Group and Type fields represent the first three qualifiers of a DSN.

For CMS, the Project and Type fields represent the FileMode and FileType tokens respectively. The Group field is ignored.

For VSE, the Project, Group and Type fields represent a LIBR library name, sub-library name and member type respectively.

`Max Entries:`

This input field (ZMAXENTRIES) specifies the maximum number of list items to be displayed in the current Favourite Datasets/Commands panel.

This number of list items may be increased or decreased at any time while the panel is open, so adding list items to or removing list from the panel's display. A command stream or fileid assigned to a list entry that has been removed, is not lost and will be redisplayed if the Max Entries value is increased to include this list item. Since panel field values are saved as SELCOPY/i User INI file variables, list item values can be redisplayed across SELCOPY/i sessions.

Keeping the maximum number of list items low, reduces the amount of storage required to display the Favourite Datasets/Commands panel and also makes the list more managable.

Default number of entries is 99, the maximum value of this field is 999.

`Scroll-H:`

Input field Scroll-H (ZSCROLLH) is used for the horizontal scrolling of text in list items. It identifies the first text position currently in view in all list items. This value may be updated by the user, so scrolling the list item entry fields so that this text position is the first in view.

Horizontal scrolling of list item text is also achieved using <PF10> (left) and <PF11> (right) from anywhere within the panel display.

Default value is 1. (i.e. the start of the list item text.)

`Columns:`

Input field Columns (ZDISPLAYCOLS) specifies the number of list item columns to be displayed on a single line of the current Favourite Datasets/Commands panel.

Increasing this value will increase the number of list items visible in the panel but will reduce the length of text displayed in each list field.

Default value is 2 columns per line.

`n:`

A number of list item (FAV.n) input field entries in which to store commonly accessed fileids and CLI command streams.

The number of list item fields displayed in the panel is defined by the Max Entries input field up to a maximum of 999.

The display of list item entries may be scrolled to a specific list item by overtyping the first item number in the current display (highlighted with red underscore) with the required list item number.

The entire contents of an individual list item may be expanded using <PF2>, allowing edit of the text data. When the expanded data view is closed (using <PF3>) the list item displays the updated text.

From anywhere within the panel display, <PF10> and <PF11> will scroll the contents of **all** list item fields left and right respectively, whereas <PF7> and <PF8> will scroll the list items up and down respectively.

System Information Menu (=8.8)

The System Information Menu panel (ZZSGSYSM) is an interactive panel window opened on selection of option 8. in the SELCOPY/i Utilities Menu.

SELCOPY/i supports display of the status information about the environment in which it is running. With the exception of operating system information windows, system windows are usually only required for diagnostic purposes only.

Option 'About' provides information about the running version of SELCOPY/i including product release and component build levels. It also identifies the latest PTF level.

Options

1 Operating System	SYSI - Operating System Details
2 LPA	SYSLPA - List LPA Modules
3 Link List	SYSLL - List Link-List libraries
4 APF Libraries	SYSAPF - List APF authorised libraries
5 Tasks	SYST - Task List
6 Programs	SYSP - List loaded programs
7 SVC Status	SVC - Display status of the SELCOPY/i SVC
8 **About**	ABOUT - Display SELCOPY/i Release and Service Level

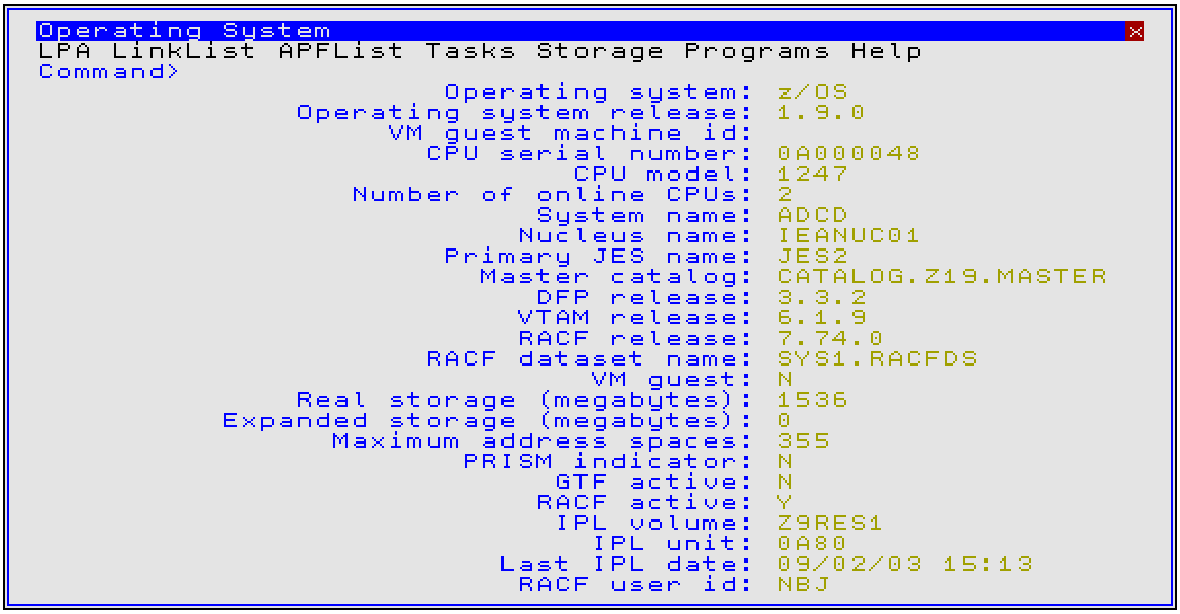
Operating System Window (=8.8.1)

The Operating System window may be opened via the following:

- Select option 1. 'Operating System' from the System Information Menu.
- Select 'Operating System' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command SYSI on the command line of any window.

Note: Not implemented for VSE and CMS.

Access to this operating system information and also to LPA, LinkList, APFList, Tasks, Storage and Programs information windows, may be restricted if RACF (or equivalent) resource checking has been enabled for the SELCOPY/i and the user does not have read access to the named SYSTEM resource.



```
Operating System
LPA LinkList APFList Tasks Storage Programs Help
Command>
                      Operating system: z/OS
              Operating system release: 1.9.0
                   VM guest machine id:
                     CPU serial number: 0A000048
                             CPU model: 1247
                 Number of online CPUs: 2
                           System name: ADCD
                          Nucleus name: IEANUC01
                      Primary JES name: JES2
                        Master catalog: CATALOG.Z19.MASTER
                           DFP release: 3.3.2
                          VTAM release: 6.1.9
                          RACF release: 7.74.0
                     RACF dataset name: SYS1.RACFDS
                              VM guest: N
              Real storage (megabytes): 1536
          Expanded storage (megabytes): 0
                Maximum address spaces: 355
                       PRISM indicator: N
                            GTF active: N
                           RACF active: Y
                            IPL volume: Z9RES1
                              IPL unit: 0A80
                         Last IPL date: 09/02/03 15:13
                          RACF user id: NBJ
```

Figure 91. MVS Operating System window.

Menu Bar Items

LPA	Open LPA Modules Window
LinkList	Open Link List Window
APFList	Open APF List Window
Tasks	Open Task List Window
Storage	Open Allocated Storage Windows
Programs	Open Loaded Programs Window

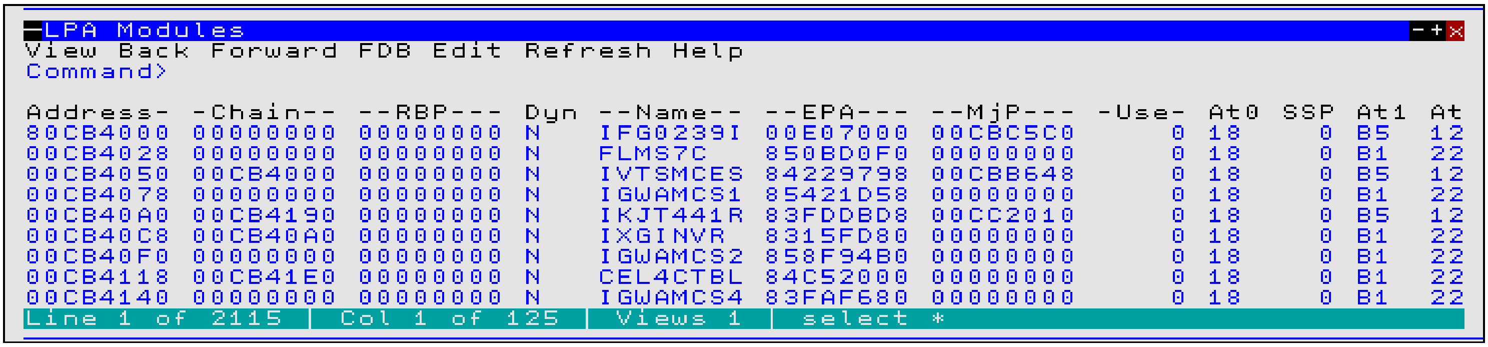
LPA Modules Window (=8.8.2)

The LPA (Link Pack Area) Modules window may be opened via the following:

- Select option 2. 'LPA' from the System Information Menu.
- Select 'LPA' from the Operating System window menu.
- Enter command SYSLPA on the command line of any window.

The LPA Modules window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not valid for CMS and VSE.



```
LPA Modules
View Back Forward FDB Edit Refresh Help
Command>

Address- -Chain-- --RBP--- Dyn --Name-- --EPA--- --MjP--- -Use- At0 SSP At1 At
80CB4000 00000000 00000000 N   IFG0239I 00E07000 00CBC5C0     0 18    0 B5  12
00CB4028 00000000 00000000 N   FLMS7C   850BD0F0 00000000     0 18    0 B1  22
00CB4050 00CB4000 00000000 N   IVTSMCES 84229798 00CBB648     0 18    0 B5  12
00CB4078 00000000 00000000 N   IGWAMCS1 85421D58 00000000     0 18    0 B1  22
00CB40A0 00CB4190 00000000 N   IKJT441R 83FDDBD8 00CC2010     0 18    0 B5  12
00CB40C8 00CB40A0 00000000 N   IXGINVR  8315FD80 00000000     0 18    0 B1  22
00CB40F0 00000000 00000000 N   IGWAMCS2 858F94B0 00000000     0 18    0 B1  22
00CB4118 00CB41E0 00000000 N   CEL4CTBL 84C52000 00000000     0 18    0 B1  22
00CB4140 00000000 00000000 N   IGWAMCS4 83FAF680 00000000     0 18    0 B1  22
Line 1 of 2115 | Col 1 of 125 | Views 1 | select *
```

Figure 92. LPA Modules window.

Columns Displayed

Name	Type	Description
Address	Hex	LPDE or CDE element address
Chain	Hex	LPDE or CDE chain pointer
RBP	Hex	RB pointer
Dyn	BitFlag	Dynamic LPA
Name	Char	Module name
EPA	Hex	Entry point address
MjP	Hex	Major name pointer
Use	UInt	Use count
At0	Hex	Attributes 0
SSP	UInt	Storage subpool
At1	Hex	Attributes 1
At2	Hex	Attributes 2
At3	Hex	Attributes 3
At4	Hex	Attributes 4
AliasOf	Char	Aliased name
LoadedAt	Hex	Load point address
LenHex	Hex	Load module length
LenDec	UInt	Load module length

Link List Window (=8.8.3)

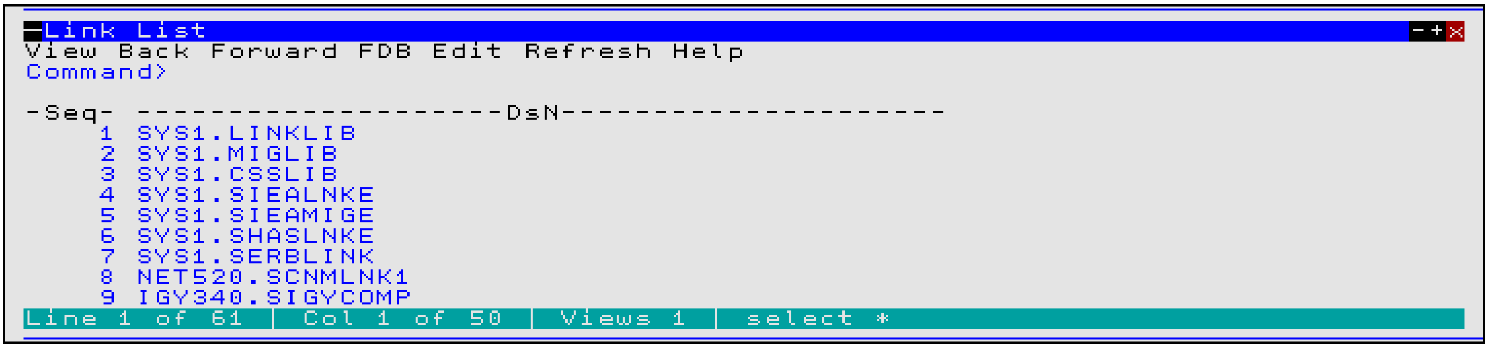
The Link List window may be opened via the following:

- Select option 3. 'Link List' from the System Information Menu.
- Select 'LinkList' from the Operating System window menu.
- Enter command SYSLL on the command line of any window.

The Link List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

The contents of the library is displayed in a List Library Members window on hitting <Enter> or, if configured, double-clicking the left mouse button, on that library's entry in the list.

Note: Not valid for CMS and VSE.



```
=Link List
View Back Forward FDB Edit Refresh Help
Command>

-Seq- --------------------DsN---------------------
     1 SYS1.LINKLIB
     2 SYS1.MIGLIB
     3 SYS1.CSSLIB
     4 SYS1.SIEALNKE
     5 SYS1.SIEAMIGE
     6 SYS1.SHASLNKE
     7 SYS1.SERBLINK
     8 NET520.SCNMLNK1
     9 IGY340.SIGYCOMP
Line 1 of 61 | Col 1 of 50 | Views 1 | select *
```

Figure 93. Link List window.

Columns Displayed

Name	Type	Description
Seq	UInt	Link list sequence number
DsN	Char	Link list library name

APF List Window (=8.8.4)

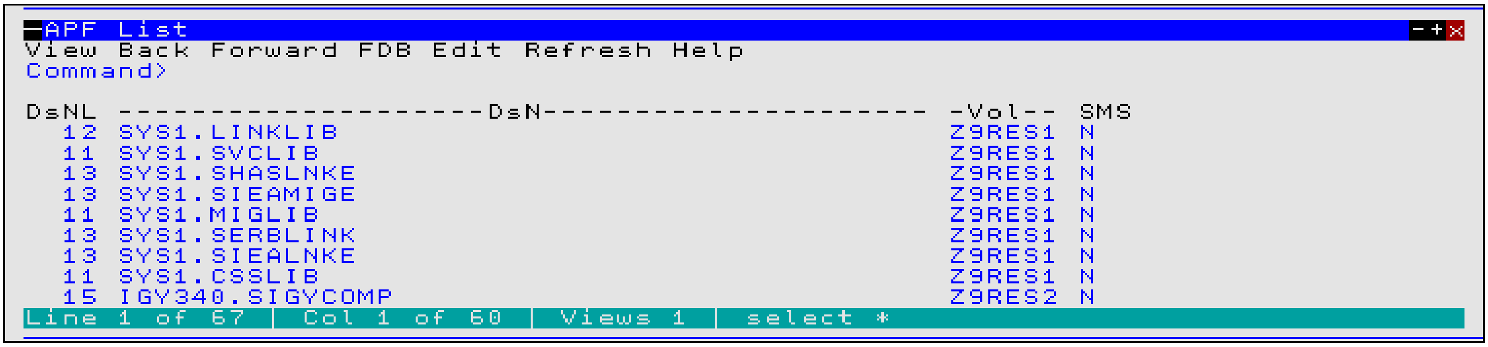
The APF (Authorised Program Facility) List window may be opened via the following:

- Select option 4. 'APF Libraries' from the System Information Menu.
- Select 'APFList' from the Operating System window menu.
- Enter command SYSAPF on the command line of any window.

The APF List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

The contents of the authorised library is displayed in a List Library Members window on hitting <Enter> or, if configured, double-clicking the left mouse button, on that library's entry in the list.

Note: Not valid for CMS and VSE.



```
 APF List                                                                    -+x
View Back Forward FDB Edit Refresh Help
Command>

DsNL --------------------DsN---------------------- -Vol-- SMS
   12 SYS1.LINKLIB                                  Z9RES1 N
   11 SYS1.SVCLIB                                   Z9RES1 N
   13 SYS1.SHASLNKE                                 Z9RES1 N
   13 SYS1.SIEAMIGE                                 Z9RES1 N
   11 SYS1.MIGLIB                                   Z9RES1 N
   13 SYS1.SERBLINK                                 Z9RES1 N
   13 SYS1.SIEALNKE                                 Z9RES1 N
   11 SYS1.CSSLIB                                   Z9RES1 N
   15 IGY340.SIGYCOMP                               Z9RES2 N
Line 1 of 67 | Col 1 of 60 | Views 1 | select *
```

Figure 94. APF List window.

Columns Displayed

Name	Type	Description
DsNL	UInt	APF library name length
DsN	Char	APF library name
Vol	Char	Volume serial
SMS	BitFlag	SMS managed

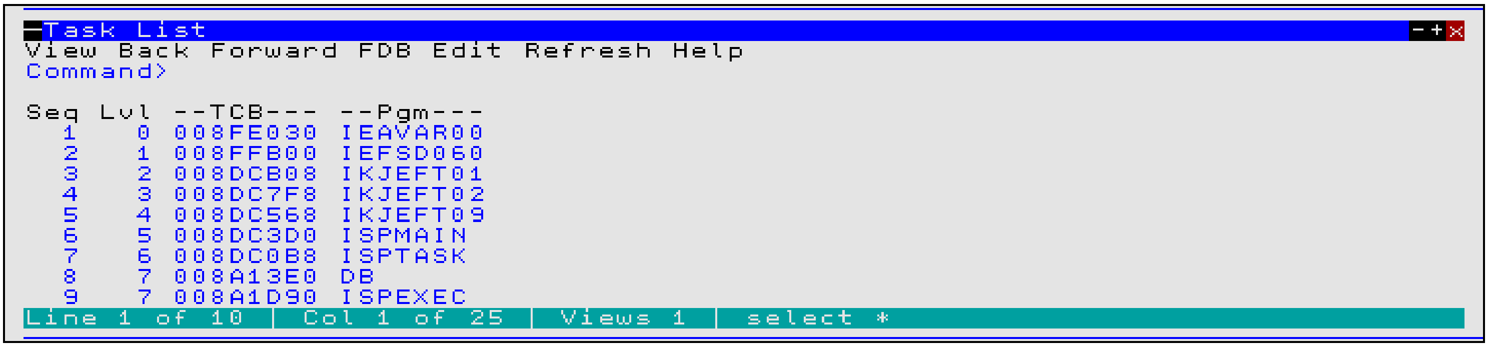
Task List Window (=8.8.5)

The Task List window may be opened to display the active tasks in the local address space, via the following:

- Select option 5. 'Tasks' from the System Information Menu.
- Select 'Tasks' from the Operating System window menu.
- Enter command SYSTASK on the command line of any window.

The Task List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not implemented for CMS and VSE.



```
 Task List                                                                   -+x
View Back Forward FDB Edit Refresh Help
Command>

Seq Lvl --TCB--- --Pgm---
  1   0 008FE030 IEAVAR00
  2   1 008FFB00 IEFSD060
  3   2 008DCB08 IKJEFT01
  4   3 008DC7F8 IKJEFT02
  5   4 008DC568 IKJEFT09
  6   5 008DC3D0 ISPMAIN
  7   6 008DC0B8 ISPTASK
  8   7 008A13E0 DB
  9   7 008A1D90 ISPEXEC
Line 1 of 10 | Col 1 of 25 | Views 1 | select *
```

Figure 95. Task List window.

Columns Displayed

Name	Type	Description
Seq	UInt	Task sequence
Lvl	UInt	Task level
TCB	Hex	TCB address
Pgm	Char	Program name

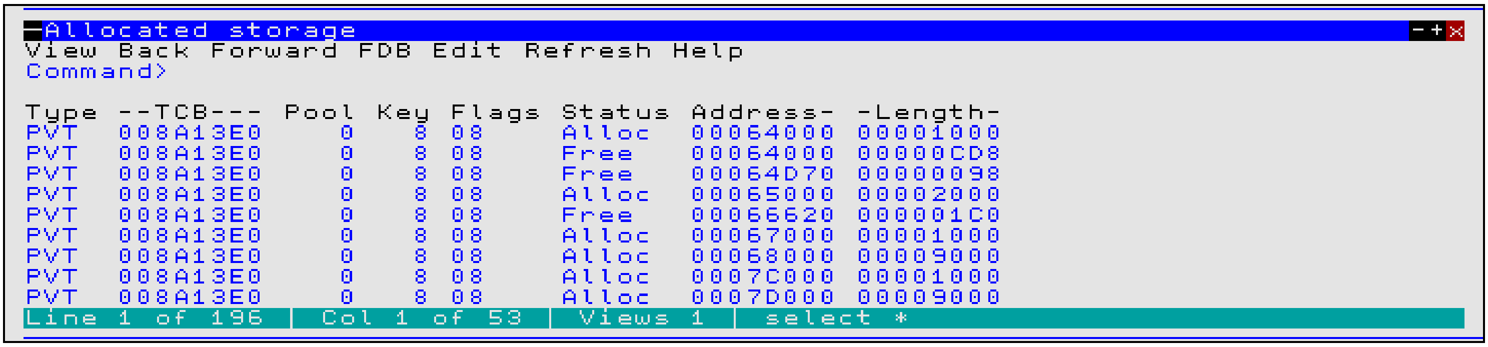
Allocated Storage Windows

Allocated Storage and Unallocated Storage windows may be opened to display lists of areas of allocated, free and unallocated storage in the local address space.

Separate storage window lists are available for Private Area (PVT), Common Service Area (CSA), System Queue Area (SQA) and Local System Queue Area (LSQA) storage. These lists may be opened by selecting the required storage type from a pop-up menu displayed on selecting 'Storage' from the Operating System window menu.

The storage windows belong to the list window class and support standard list window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not implemented for CMS and VSE.



```
Allocated storage
View Back Forward FDB Edit Refresh Help
Command>

Type  --TCB---  Pool  Key  Flags  Status  Address-  -Length-
PVT   008A13E0     0    8  08     Alloc   00064000  00001000
PVT   008A13E0     0    8  08     Free    00064000  00000CD8
PVT   008A13E0     0    8  08     Free    00064D70  00000098
PVT   008A13E0     0    8  08     Alloc   00065000  00002000
PVT   008A13E0     0    8  08     Free    00066620  000001C0
PVT   008A13E0     0    8  08     Alloc   00067000  00001000
PVT   008A13E0     0    8  08     Alloc   00068000  00009000
PVT   008A13E0     0    8  08     Alloc   0007C000  00001000
PVT   008A13E0     0    8  08     Alloc   0007D000  00009000
Line 1 of 196 | Col 1 of 53 | Views 1 | select *
```

Figure 96. Allocated Private Storage window.

Columns Displayed

Name	Type	Description
Type	Char	Storage type
TCB	Hex	Owning TCB
Pool	UInt	Sub pool
Key	UInt	Sub pool storage key
Flags	Hex	Sub pool flags
Status	Char	Storage status
Address	Hex	Storage address
Length	Hex	Storage length

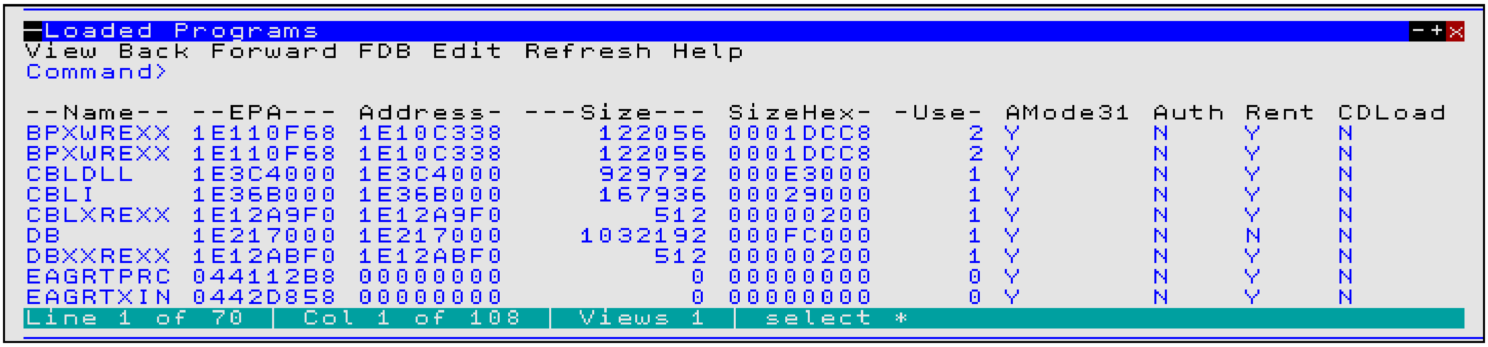
Loaded Programs Window (=8.8.6)

The Loaded Programs window may be opened to display programs that have been dynamically loaded into the local address space, via the following:

- Select option 6. 'Programs' from the System Information Menu.
- Select 'Programs' from the Operating System window menu.
- Enter command SYSPGM on the command line of any window.

The Loaded Programs window is a List Window and supports standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not implemented for CMS and VSE.



```
 Loaded Programs
View Back Forward FDB Edit Refresh Help
Command>

--Name-- --EPA--- Address- ---Size--- SizeHex- -Use- AMode31 Auth Rent CDLoad
BPXWREXX 1E110F68 1E10C338     122056 0001DCC8     2 Y       N    Y    N
BPXWREXX 1E110F68 1E10C338     122056 0001DCC8     2 Y       N    Y    N
CBLDLL   1E3C4000 1E3C4000     929792 000E3000     1 Y       N    Y    N
CBLI     1E36B000 1E36B000     167936 00029000     1 Y       N    Y    N
CBLXREXX 1E12A9F0 1E12A9F0        512 00000200     1 Y       N    Y    N
DB       1E217000 1E217000    1032192 000FC000     1 Y       N    N    N
DBXXREXX 1E12ABF0 1E12ABF0        512 00000200     1 Y       N    Y    N
EAGRTPRC 044112B8 00000000          0 00000000     0 Y       N    Y    N
EAGRTXIN 0442D858 00000000          0 00000000     0 Y       N    Y    N
Line 1 of 70 | Col 1 of 108 | Views 1 | select *
```

Figure 97. Loaded Programs window.

Columns Displayed

Name	Type	Description
Name	Char	Program name.
EPA	Hex	Entry point address.
Address	Hex	Load address.
Size	UInt	Load module size.
SizeHex	Hex	Load module size (hex).
Use	UInt	Use count.
AMode31	BitFlag	Program is AMODE 31.
Auth	BitFlag	Program is authorised.
Rent	BitFlag	Program is reuseable.
CDLoad	BitFlag	Program loaded with VSE CDLOAD.
Perm	BitFlag	CMS nucleus extension PERM attribute.
Sys	BitFlag	CMS nucleus extension SYSTEM attribute.
Service	BitFlag	CMS nucleus extension SERVICE attribute.
EndCmd	BitFlag	CMS nucleus extension ENDCMD attribute.
ImmCmd	BitFlag	CMS nucleus extension IMMCMD attribute.

SELCOPY/i Storage Statistics Window

The Storage Statistics window may be opened via the following:

- Select 'SELCOPY/i storage stats' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command SYSSTOR on the command line of any window.

The Storage Statistics window displays storage being used by SELCOPY/i at that moment in time. The values in each field will vary as windows are opened and closed.

The storage allocated by SELCOPY/i is categorised internally as belonging to the Heap or the Stack.

Heap

Heap storage contains structures such as lists, control blocks, etc.

Each structure is an element within the heap and may persist beyond the life of the function that generated it. Each element exists within a fixed length heap storage block which itself may contain 1 or more elements. When an element is released by SELCOPY/i, the area within the storage block occupied by that element, is freed.

If possible, SELCOPY/i will utilise these free areas of storage for new elements. However, if an element is generated with a length that exceeds the available free area within existing storage blocks, a new storage block is allocated from main storage. Similarly, if all the elements within a storage block are freed, the block is released back to main storage.

Stack

The stack is a fixed area of storage that contains the dynamic storage areas associated with each function that has been called.

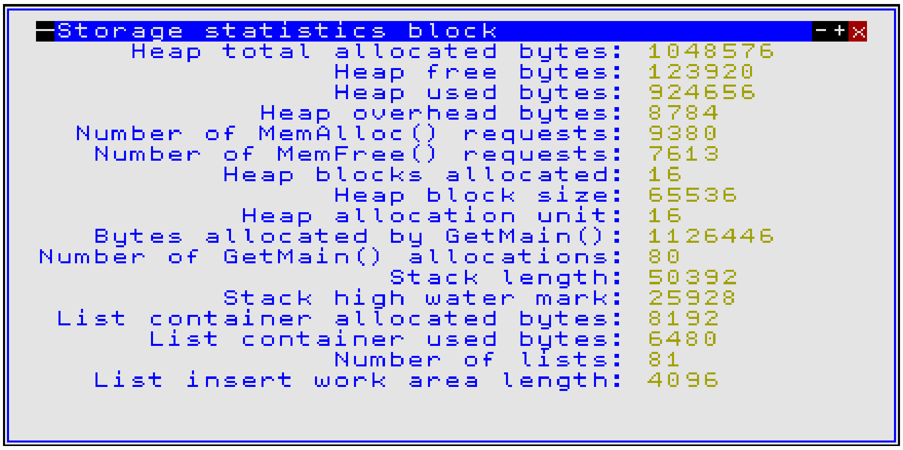
A function acquires dynamic storage for local variables from the stack when it is called and frees this storage when it returns control to its caller.

Thus, the amount of stack storage in use at any time depends on the number of levels of nested function calls and the amount of storage required by each function. This, in turn, depends on which facilities of SELCOPY/i are in use.

The **Stack high water mark** represents the maximum amount of stack storage that has been in use in the current SELCOPY/i session.

Lists

The List insert work area and List container are areas of storage associated with SELCOPY/i listing facilities.



```
Storage statistics block
    Heap total allocated bytes: 1048576
               Heap free bytes: 123920
               Heap used bytes: 924656
           Heap overhead bytes: 8784
 Number of MemAlloc() requests: 9380
  Number of MemFree() requests: 7613
         Heap blocks allocated: 16
               Heap block size: 65536
          Heap allocation unit: 16
  Bytes allocated by GetMain(): 1126446
Number of GetMain() allocations: 80
                  Stack length: 50392
         Stack high water mark: 25928
 List container allocated bytes: 8192
      List container used bytes: 6480
               Number of lists: 81
  List insert work area length: 4096
```

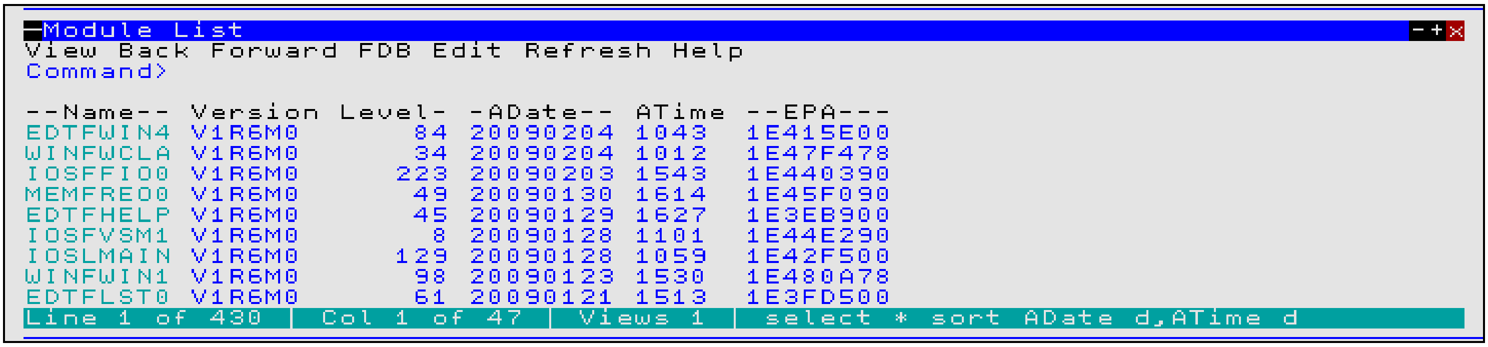
Figure 98. Storage Statistics window.

SELCOPY/i Module List Window

The SELCOPY/i Module List may be opened to display information on all modules that comprise SELCOPY/i, via the following:

- Select 'SELCOPY/i module list' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command APE on the command line of any window.

The Module List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.



```
Module List
View Back Forward FDB Edit Refresh Help
Command>

--Name-- Version Level- -ADate-- ATime --EPA---
EDTFWIN4 V1R6M0      84 20090204 1043  1E415E00
WINFWCLA V1R6M0      34 20090204 1012  1E47F478
IOSFFI00 V1R6M0     223 20090203 1543  1E440390
MEMFRE00 V1R6M0      49 20090130 1614  1E45F090
EDTFHELP V1R6M0      45 20090129 1627  1E3EB900
IOSFVSM1 V1R6M0       8 20090128 1101  1E44E290
IOSLMAIN V1R6M0     129 20090128 1059  1E42F500
WINFWIN1 V1R6M0      98 20090123 1530  1E480A78
EDTFLST0 V1R6M0      61 20090121 1513  1E3FD500
Line 1 of 430 | Col 1 of 47 | Views 1 | select * sort ADate d,ATime d
```

Figure 99. Module List window.

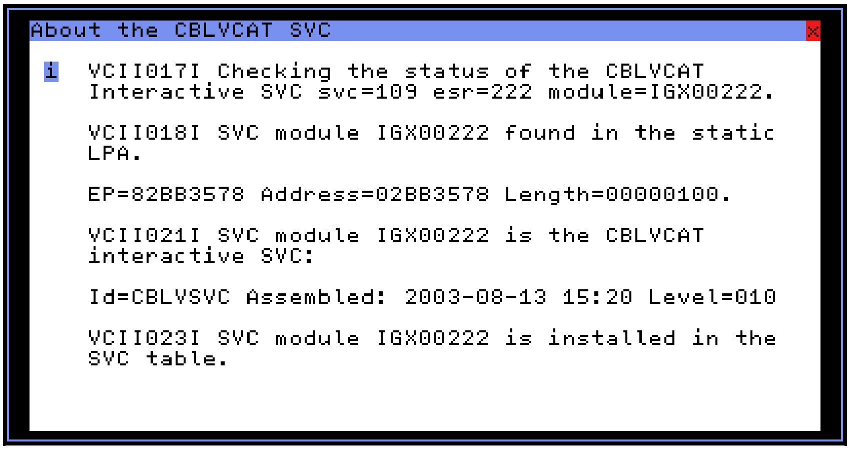
CBLVCAT SVC window (=8.8.7)

The CBLVCAT SVC window may be opened via the following:

- Select option 7. 'SVC Status' from the System Information Menu.
- Select 'SELCOPY/i SVC' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command SVC on the command line of any window.

The CBLVCAT SVC window displays information about the CBLVCAT SVC required to perform CBLVCAT LISTVCAT catalog listings.

Note: Not valid for CMS and VSE.



```
About the CBLVCAT SVC

i  VCII017I Checking the status of the CBLVCAT
   Interactive SVC svc=109 esr=222 module=IGX00222.

   VCII018I SVC module IGX00222 found in the static
   LPA.

   EP=82BB3578 Address=02BB3578 Length=00000100.

   VCII021I SVC module IGX00222 is the CBLVCAT
   interactive SVC:

   Id=CBLVSVC Assembled: 2003-08-13 15:20 Level=010

   VCII023I SVC module IGX00222 is installed in the
   SVC table.
```

Figure 100. CBLVCAT SVC window.

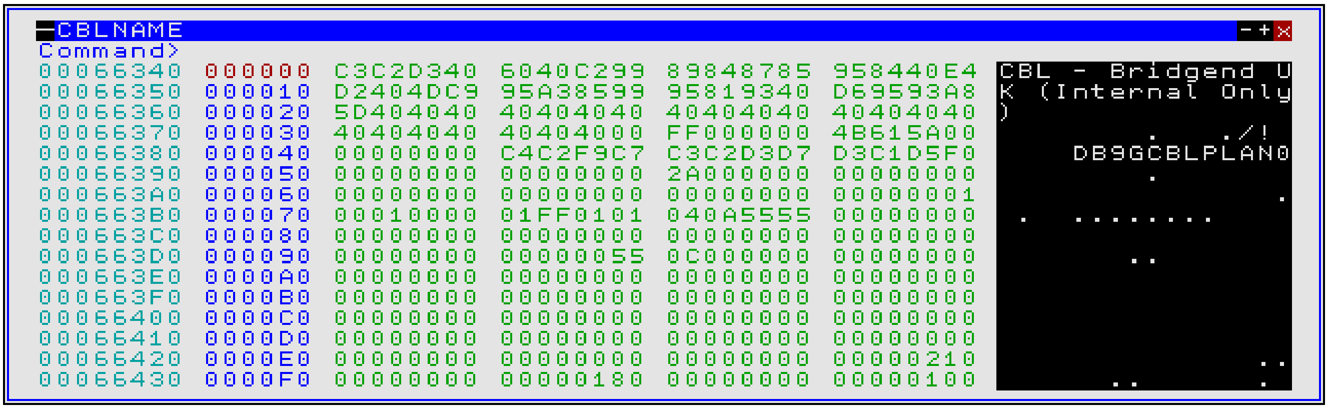
CBLNAME Window

The CBLVCAT SVC window may be opened via the following:

- Select 'CBLNAME' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command CBLNAME on the command line of any window.

The CBLNAME window is a storage display window containing the CBLNAME module loaded by SELCOPY/i.

The CBLNAME storage display window does not display areas of storage outside the loaded CBLNAME module and the data may not be updated by the user.



```
CBLNAME
Command>
00066340 000000 C3C2D340 6040C299 89848785 958440E4 CBL - Bridgend U
00066350 000010 D2404DC9 95A38599 95819340 D69593A8 K (Internal Only
00066360 000020 5D404040 40404040 40404040 40404040 )
00066370 000030 40404040 40404000 FF000000 4B615A00         .   ./!
00066380 000040 00000000 C4C2F9C7 C3C2D3D7 D3C1D5F0     DB9GCBLPLAN0
00066390 000050 00000000 00000000 2A000000 00000000         .
000663A0 000060 00000000 00000000 00000000 00000001                .
000663B0 000070 00010000 01FF0101 040A5555 00000000  .  ........
000663C0 000080 00000000 00000000 00000000 00000000
000663D0 000090 00000000 00000055 0C000000 00000000        ..
000663E0 0000A0 00000000 00000000 00000000 00000000
000663F0 0000B0 00000000 00000000 00000000 00000000
00066400 0000C0 00000000 00000000 00000000 00000000
00066410 0000D0 00000000 00000000 00000000 00000000
00066420 0000E0 00000000 00000000 00000000 00000210               ..
00066430 0000F0 00000000 00000180 00000000 00000100       ..      .
```

Figure 101. CBLNAME window.

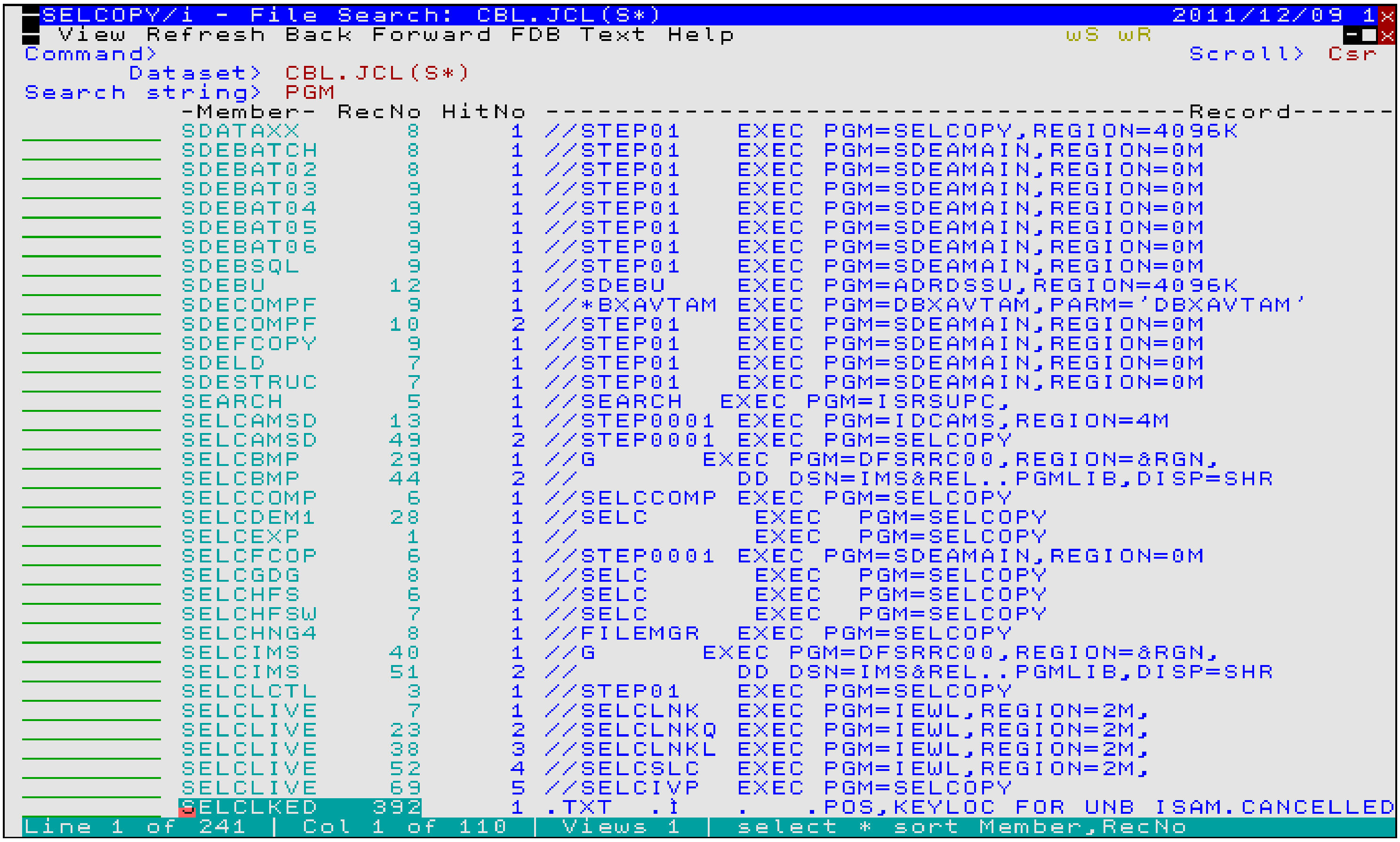
File Search (=8.9)

The File Search window may be opened via the following:

- Select option 9. 'Search' from the Utilities Menu.
- Select 'File Search' from the Utilities menu in the CBLe main window menu bar.
- Enter command FS on the command line of any window.

The File Search window displays the lines in a PDS member (MVS), LIBR member (VSE) or CMS file that contain a given string.

For more advanced file search features, use the File Search/Update/Copy utility.



```
 SELCOPY/i - File Search: CBL.JCL(S*)                             2011/12/09 1
  View Refresh Back Forward FDB Text Help                         wS wR
Command>                                                             Scroll> Csr
      Dataset> CBL.JCL(S*)
Search string> PGM
         -Member- RecNo HitNo ------------------------------------Record------
         SDATAXX      8     1 //STEP01   EXEC PGM=SELCOPY,REGION=4096K
         SDEBATCH     8     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBAT02     8     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBAT03     9     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBAT04     9     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBAT05     9     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBAT06     9     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBSQL      9     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEBU       12     1 //SDEBU    EXEC PGM=ADRDSSU,REGION=4096K
         SDECOMPF     9     1 //*BXAVTAM EXEC PGM=DBXAVTAM,PARM='DBXAVTAM'
         SDECOMPF    10     2 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDEFCOPY     9     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDELD        7     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SDESTRUC     7     1 //STEP01   EXEC PGM=SDEAMAIN,REGION=0M
         SEARCH       5     1 //SEARCH  EXEC PGM=ISRSUPC,
         SELCAMSD    13     1 //STEP0001 EXEC PGM=IDCAMS,REGION=4M
         SELCAMSD    49     2 //STEP0001 EXEC PGM=SELCOPY
         SELCBMP     29     1 //G       EXEC PGM=DFSRRC00,REGION=&RGN,
         SELCBMP     44     2 //         DD  DSN=IMS&REL..PGMLIB,DISP=SHR
         SELCCOMP     6     1 //SELCCOMP EXEC PGM=SELCOPY
         SELCDEM1    28     1 //SELC      EXEC  PGM=SELCOPY
         SELCEXP      1     1 //          EXEC  PGM=SELCOPY
         SELCFCOP     6     1 //STEP0001 EXEC PGM=SDEAMAIN,REGION=0M
         SELCGDG      8     1 //SELC      EXEC  PGM=SELCOPY
         SELCHFS      6     1 //SELC      EXEC  PGM=SELCOPY
         SELCHFSW     7     1 //SELC      EXEC  PGM=SELCOPY
         SELCHNG4     8     1 //FILEMGR  EXEC PGM=SELCOPY
         SELCIMS     40     1 //G       EXEC PGM=DFSRRC00,REGION=&RGN,
         SELCIMS     51     2 //         DD  DSN=IMS&REL..PGMLIB,DISP=SHR
         SELCLCTL     3     1 //STEP01   EXEC PGM=SELCOPY
         SELCLIVE     7     1 //SELCLNK  EXEC PGM=IEWL,REGION=2M,
         SELCLIVE    23     2 //SELCLNKQ EXEC PGM=IEWL,REGION=2M,
         SELCLIVE    38     3 //SELCLNKL EXEC PGM=IEWL,REGION=2M,
         SELCLIVE    52     4 //SELCSLC  EXEC PGM=IEWL,REGION=2M,
         SELCLIVE    69     5 //SELCIVP  EXEC PGM=SELCOPY
         SELCLKED   392     1 .TXT   .İ    .    .POS,KEYLOC FOR UNB ISAM.CANCELLED
Line 1 of 241 | Col 1 of 110 | Views 1 | select * sort Member,RecNo
```

Figure 102. File Search window.

Panel Fields

`Dataset>`

◊ For MVS, the Dataset parameter is the DSN of a PDS(E) library to be searched which may optionally include a member name mask to identify a subset of members to be searched.

A member name mask supports the following wild cards:

`*`	A single asterisk represents an entire member name or zero or more characters within a member name mask.
`%`	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS(E) DSN and be enclosed in "( )" (parentheses). A member name mask that is less than 8 characters in length and does not contain an "*" (asterisk) wild card will have a trailing "*" wild card automatically appended. e.g. To search all members of "CBP.PGMLIB" whose names start "CBLA":

```
CBL.PGMLIB(CBLA)
```

◊ For VSE, the Dataset parameter is the name of the LIBR library and sub-library to be searched. The sub-library name, member name and member type may include the "*" wild card to represent zero or more characters. e.g. To search all members of "OEM2.CBL" :

```
OEM2.CBL.*.*
```

◊ For CMS, the Dataset parameter is a CMS fileid mask in standard CMS format denoting the files to be searched. The file name, file type and file mode may each include the "*" wild card to represent zero or more characters. e.g. To search all "EXEC" file types with file name beginning "SS" on all accessed mini-disks.

```
SS* EXEC *
```

`Search string>`

The character search string.

The search string is **not** case sensitive and must be enclosed in single or double quotes if it includes blank characteres.

Prefix Line Commands

For **MVS** systems, the following prefix line commands are available:

Command	**Description**
<Dflt>	Prefix line command E.
A	Open the Create Alias dialog window.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EX	Execute the entry. (Invokes the TSO command, EXECUTE, using the entry name as input.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FS	Open the File Search window for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

For **VSE** systems, the following prefix line commands are available:

Command	**Description**
<Dflt>	Prefix line command E.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
FS	Open the File Search window to search the contents of this entry. Not supported for VSE LIBR library entries.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
L	LOCK the member.
R	Rename the entry.
U	UNLOCK the member. A member may only be unlocked by the user that locked it.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command.

	Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Columns Displayed

Name	**Type**	**Description**
Member	ALPair	Member
RecNo	Int	Record number
HitNo	Int	Hit number
Record	ALPair	File record

Search for Library Members (=8.10)

Overview

The Search for Library Members utility (LLX) provides a method of locating by name, one or more members within a number of PDS/PDSE libraries.

The utility calls the LL (ListLibrary) command repeatedly for each library with a DSN matching the library DSN mask(s) or referenced by a DDname or DDname concatenation of libraries. This will identify the existance of any members that match the specified member name mask(s) within that library.

The Search for Library Members utility executes in the foreground only. To execute in batch, a user can write a simple SELCOPY routine to read the libraries with parameter DIR and report the library directory entries that match the required member mask. See sample SELCOPY routine members (ZZI*) in sample library SZZSSAM1 and also *"SELCOPY Debug & Development"* for assistance in writing new SELCOPY routines.

The Search for Library Members report output is written to a temporary file and presented to the user in a SELCOPY/i (CBLe) text edit view.

Unless "Quiet" option has been selected, the library member name search will pause and a popup window opened prompting the user for a decision to cancel or continue, with or without further prompts, if either of the following conditions are true:

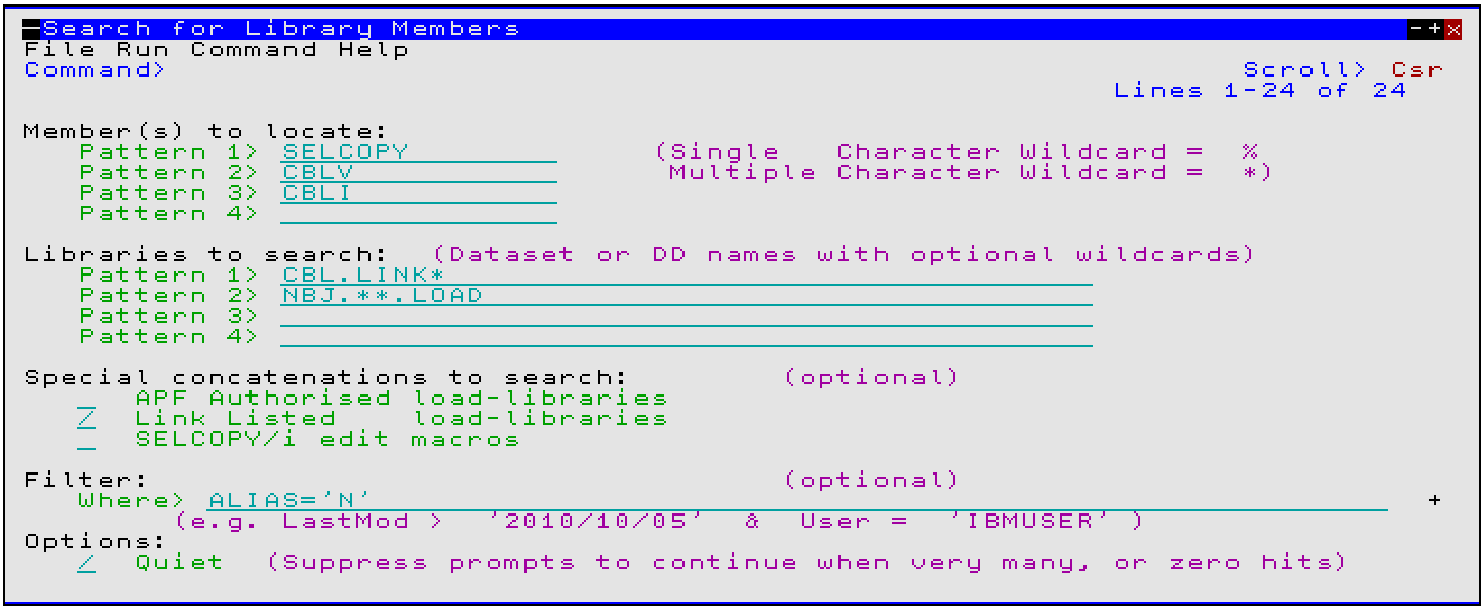
1. No matching member names have been found in the first 10 libraries. The popup window proivides an opportunity to change the number of libraries to search before this popup is displayed again.
2. A single library contains at least 1000 matching member names. The popup window proivides an opportunity to change this matching member names threshold.

Search for Library Members Panel

The Search for Library Members utility panel window (ZZSLLX00) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select option 10. 'Find Lib Members(s)' from the Utilities Menu.
- Select 'Search for Library members' from the Utilities menu.
- Execute the command LLX with no parameters from the command line of any window.

By default, field entries are populated with arguments and options that were entered the last time the Search for Library Members Utility panel was used.



```
 Search for Library Members                                          -+x
 File Run Command Help
 Command>                                                     Scroll> Csr
                                                        Lines 1-24 of 24

 Member(s) to locate:
    Pattern 1> SELCOPY              (Single   Character Wildcard =  %
    Pattern 2> CBLV                  Multiple Character Wildcard =  *)
    Pattern 3> CBLI
    Pattern 4>

 Libraries to search:  (Dataset or DD names with optional wildcards)
    Pattern 1> CBL.LINK*
    Pattern 2> NBJ.**.LOAD
    Pattern 3>
    Pattern 4>

 Special concatenations to search:          (optional)
        APF Authorised load-libraries
    /   Link Listed    load-libraries
    _   SELCOPY/i edit macros

 Filter:                                    (optional)
    Where> ALIAS='N'                                                    +
          (e.g. LastMod >  '2010/10/05'  &  User =  'IBMUSER' )
 Options:
    /   Quiet  (Suppress prompts to continue when very many, or zero hits)
```

Figure 103. Search for Library Members Panel.

Having typed entries in the required panel fields, simply pressing the <Enter> key or, if configured, double-clicking the left mouse button will will action the library member name search in the foreground.

Menu Bar Items

Run

Run the library member name search in the foreground.

Command

Generate the LLX command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using CMDTEXT point-and-shoot execution <PF4> or copied into the user's HOME file and saved for future execution.

Panel Input Fields

Members(s) to locate:
Pattern 1>
Pattern 2>
Pattern 3>
Pattern 4>

These input fields (MEMBER1, MEMBER2, MEMBER3 and MEMBER4) allow the user to provide up to 4 alternative member name masks to identify the member names to be located.

A member name mask supports the following wild cards:

*	A single asterisk represents an entire member name or zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If no member name masks are specified, then all libraries selected will be searched for all members. i.e. All members will be reported for each library.

Libraries to search:
Pattern 1>
Pattern 2>
Pattern 3>
Pattern 4>

These input fields (LIBRARY1, LIBRARY2, LIBRARY3 and LIBRARY4) allow the user to provide up to 4 alternative library DSN masks, library DDnames and/or library concatenation DDnames which identiy the PDS/PDSE libraries in which to search.

Note that **all** libraries referenced within a DDname data set concatenation will be searched.

A library DSN mask supports the following wild cards:

`*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.

`**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.

`%` A single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

If no library DSN masks or library DDnames are specified, then at least one of the special library concatenation fields (APF Authorised, Link Listed or SELCOPY/i edit macros) must be selected.

`APF Authorised load-libraries`

This option field (APFCHK) indicates that all APF authorised load libraries are to be searched.

This list of libraries may be displayed using the APF List Window (command SYSAPF.)

`Link Listed load-libraries`

This option field (LNKCHK) indicates that all load libraries in the active Link List concatenation are to be searched.

This list of libraries may be displayed using the Link List Window (command SYSLL.)

`SELCOPY/i edit macros`

This option field (MACCHK) indicates that all libraries in the user's current CBLe text editor macro path are to be searched.

This list of libraries may be displayed using the CBLe text edit command, QUERY MACROPATH.)

`Where>`

This input field (WHERE) specifies additional member name filter criteria. Members are reported as being found only if the information in the PDS/PDSE directory entry for that member also satisfies this additional criteria.

The syntax of a WHERE filter is described by the list window WHERE Clause which supports list field names as described by the MVS load library and non-load library lists' field descriptor block (FDB). See *"List Library Members"* for details of these field names, descriptions and their data types.

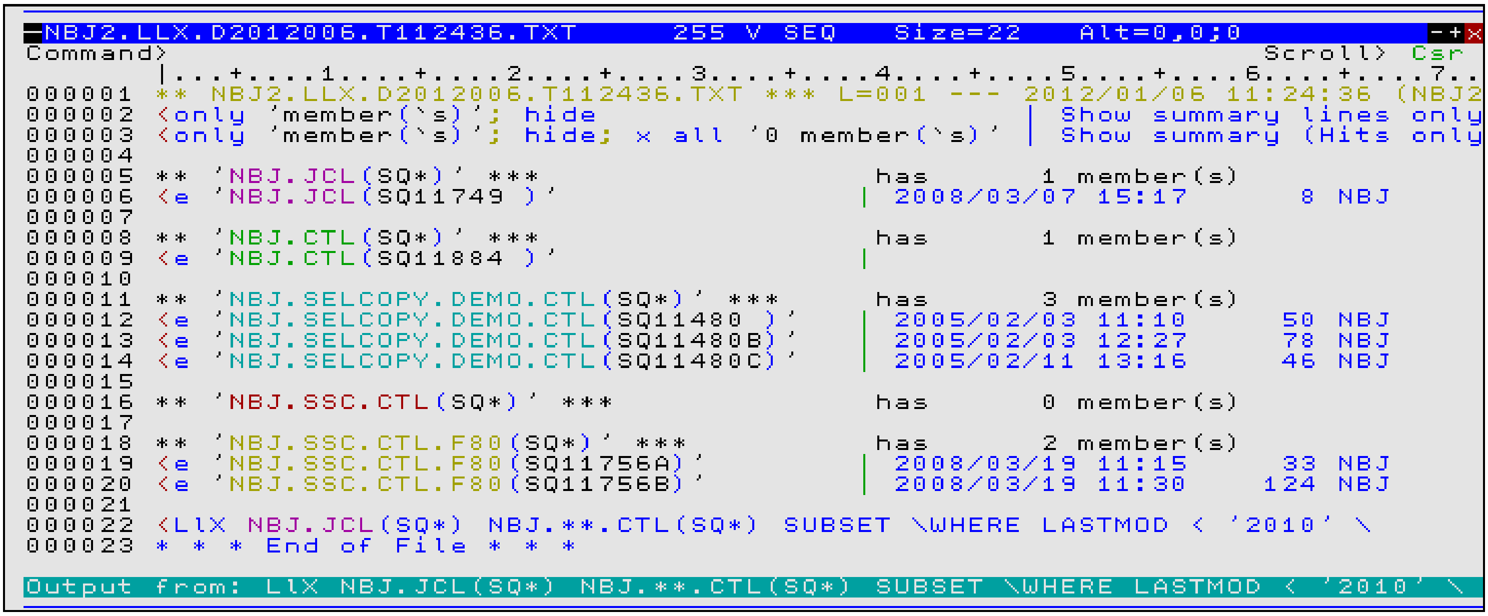
Beware that maintenance of a non-load library member's directory information is **not** enforced by the system. Therefore, its existance depends on the last application to write data to that member. Missing directory fields have default values: 0 if numeric (e.g. VV, MM); ' ' (blank) if character (e.g. User) and null if TimeDec (e.g. Created, LastMod.)

`Quiet`

This option field (QUIET) indicates that the user will not be prompted for a decision to continue the search when library or member name thresholds are encountered.

Search for Library Members Output

The output generated by the Search for Library Members utility is a temporary CMX command file which gets displayed automatically in a SELCOPY/i (CBLe) text edit window view. It identifies each of the library names searched, the number of matching member names in the library, followed by a command to edit each member located in that library.



```
-NBJ2.LLX.D2012006.T112436.TXT     255 V SEQ   Size=22   Alt=0,0;0          -+x
Command>                                                           Scroll> Csr
       |...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000001 ** NBJ2.LLX.D2012006.T112436.TXT *** L=001 --- 2012/01/06 11:24:36 (NBJ2
000002 <only 'member(`s)'; hide                       | Show summary lines only
000003 <only 'member(`s)'; hide; x all '0 member(`s)' | Show summary (Hits only
000004
000005 ** 'NBJ.JCL(SQ*)' ***                  has      1 member(s)
000006 <e 'NBJ.JCL(SQ11749 )'                | 2008/03/07 15:17      8 NBJ
000007
000008 ** 'NBJ.CTL(SQ*)' ***                  has      1 member(s)
000009 <e 'NBJ.CTL(SQ11884 )'                |
000010
000011 ** 'NBJ.SELCOPY.DEMO.CTL(SQ*)' ***     has      3 member(s)
000012 <e 'NBJ.SELCOPY.DEMO.CTL(SQ11480 )'   | 2005/02/03 11:10     50 NBJ
000013 <e 'NBJ.SELCOPY.DEMO.CTL(SQ11480B)'   | 2005/02/03 12:27     78 NBJ
000014 <e 'NBJ.SELCOPY.DEMO.CTL(SQ11480C)'   | 2005/02/11 13:16     46 NBJ
000015
000016 ** 'NBJ.SSC.CTL(SQ*)' ***              has      0 member(s)
000017
000018 ** 'NBJ.SSC.CTL.F80(SQ*)' ***          has      2 member(s)
000019 <e 'NBJ.SSC.CTL.F80(SQ11756A)'        | 2008/03/19 11:15     33 NBJ
000020 <e 'NBJ.SSC.CTL.F80(SQ11756B)'        | 2008/03/19 11:30    124 NBJ
000021
000022 <LlX NBJ.JCL(SQ*) NBJ.**.CTL(SQ*) SUBSET \WHERE LASTMOD < '2010' \
000023 * * * End of File * * *

Output from: LlX NBJ.JCL(SQ*) NBJ.**.CTL(SQ*) SUBSET \WHERE LASTMOD < '2010' \
```

Figure 104. Search for Library Members Output.

The ONLY text edit macro invocations in lines 2 and 3 may be executed to filter edited lines to display all summary lines (library DSN and number of matching member names) or only summary lines for which at leasty one member name has been found.

The last line of the file is the LLX command line syntax generated by the Search for Library Members utility for the options provided.

Any command (prefixed by <) in this file may be executed using the CMDTEXT facility simply by positioning the cursor on the required command and pressing the <PF4> key.

Calendar Window (=8.13)

The Calendar window may be opened via the following:

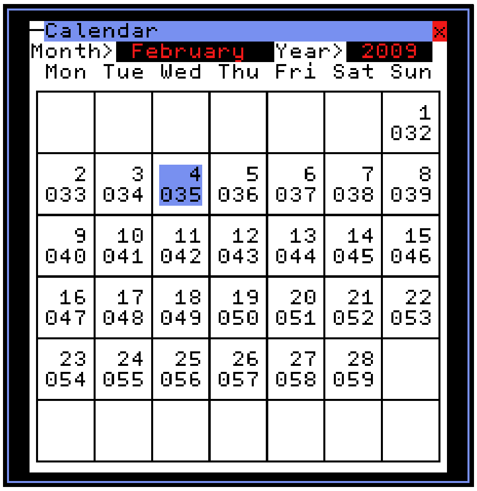
- Select option 13. 'Calendar' from the Utilities Menu.
- Select 'Calendar' from the Utilities menu in the CBLe main window menu bar.
- Enter command CALendar on the command line of any window.

When opened, the calendar window shows the current month with today's date highlighted. Each day has the day of the month and the Julian day number displayed in a table.

You can scroll the calendar backwards and forwards by the month or the year or you can enter a specific year or month in the fields at the top of the window.

To scroll the calendar use the following commands:

Command	Default PF key	Description
SCROLL UP	PF7	Display the previous month.
SCROLL DOWN	PF8	Display the next month.
SCROLL LEFT	PF11	Display the current month in the previous year.
SCROLL RIGHT	PF12	Display the current month in the next year.



Calendar

Month> February Year> 2009

Mon	Tue	Wed	Thu	Fri	Sat	Sun
						1 032
2 033	3 034	4 035	5 036	6 037	7 038	8 039
9 040	10 041	11 042	12 043	13 044	14 045	15 046
16 047	17 048	18 049	19 050	20 051	21 052	22 053
23 054	24 055	25 056	26 057	27 058	28 059	

Figure 105. Calendar window.

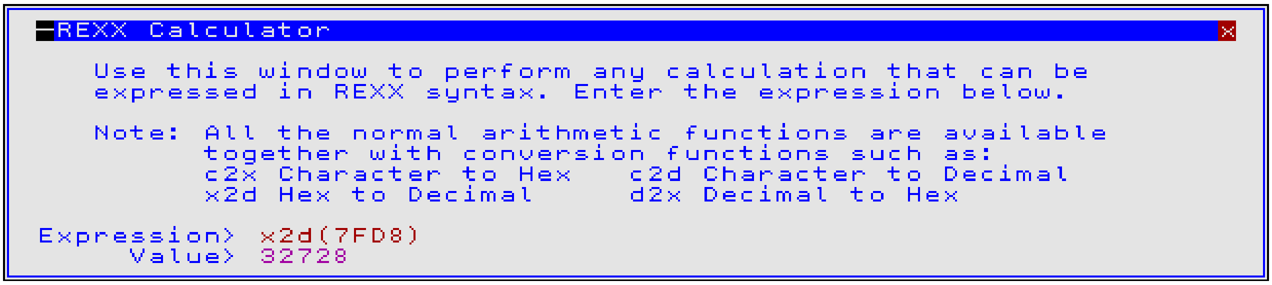
Calculator Window (=8.14)

The Calculator window may be opened via the following:

- Select option 14. 'Calculator' from the Utilities Menu.
- Select 'Calculator' from the Utilities menu in the CBLe main window menu bar.
- Enter command CALC on the command line of any window.

The calculator window allows you to enter a calculation and displays the result of the calculation.

In fact the calculator is a REXX function interpreter. You enter a valid REXX expression and the calculator evaluates it. You are not restricted to numerical calculations. You can enter any valid REXX expression including for example the conversion functions.



```
REXX Calculator

   Use this window to perform any calculation that can be
   expressed in REXX syntax. Enter the expression below.

   Note: All the normal arithmetic functions are available
         together with conversion functions such as:
         c2x Character to Hex    c2d Character to Decimal
         x2d Hex to Decimal      d2x Decimal to Hex

Expression> x2d(7FD8)
     Value> 32728
```

Figure 106. REXX Calculator window.

Create Structure (SDO) Menu (=9)

The Create Structure (SDO) Menu (ZZSGSDO0) is an interactive panel window, opened on selection of option 9. in the SELCOPY/i Primary option menu.

SELCOPY/i supports browse, edit, compare, search, update and remap of structured data records. To accurately format these records, the SELCOPY/i structured data editor uses an internally defined structure (SDO) which may be pre-generated from any combination of the following sources:

1. COBOL copybook.
2. PL1 copybook (include file).
3. COBOL or PL1 ADATA output file.
4. SELCOPY/i SDO structure file.
5. SDE CREATE STRUCTURE Record Type Definition syntax.

Additionally, the ZZSXREF utility may be used to generate a SELCOPY/i SDO from an XREF file.

This panel allows the user to select the source from which the Structured Data Edit (SDE) structure object (SDO) is to be created. Enter the relevant option number or position the cursor on the required option and press <Enter> or, if configured, double-click the left mouse button.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel.

Help

Open the general help for the Create Structure (SDO) Menu option menu panel.

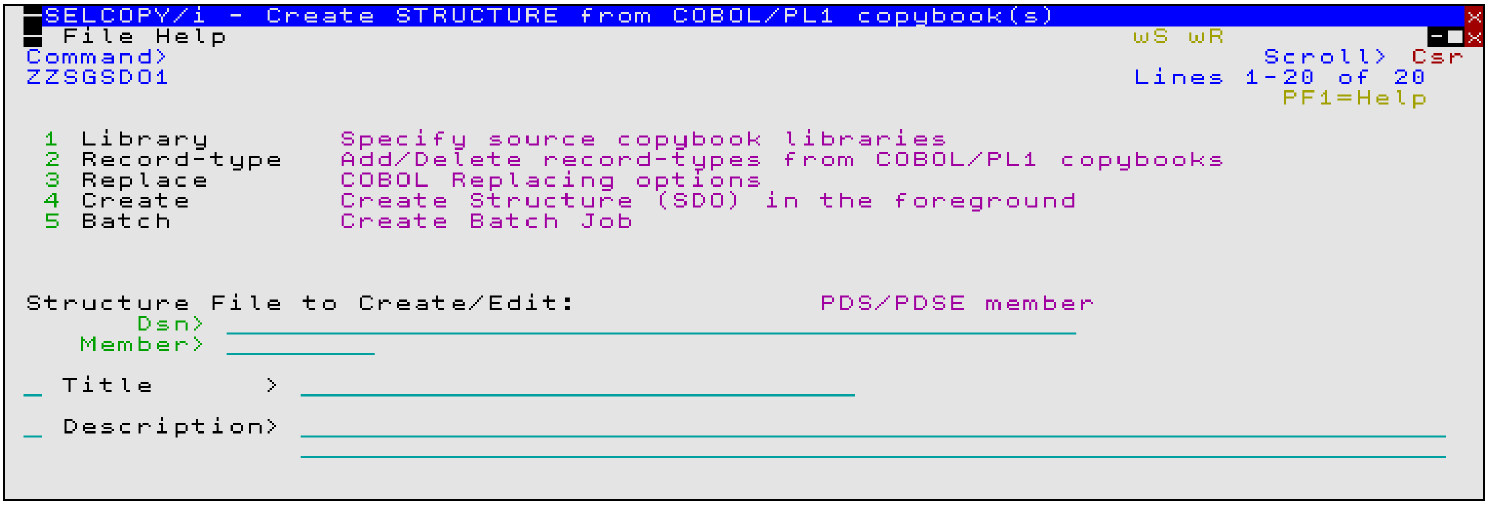
Options

1 Copybook	SDO - Create a structure from COBOL or PL1 copybook(s)
2 XREF	XREF - Create a structure from an existing XREF file
3 Layout	LAYOUT - Display record layouts defined by an SDO/copybook.

Create Structure from Copybook(s)

The Create Structure from COBOL or PL1 copybooks panels assist with generation of a SELCOPY/i structure definition file using one or more COBOL GROUP and/or PL1 STRUCTURE definitions. Each group/structure generates a single record type (RTO) mapping within the SDO.

Create Structure from COBOL/PL1 copybook(s)



```
 SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
  File Help                                                 wS wR
Command>                                                           Scroll> Csr
ZZSGSDO1                                                        Lines 1-20 of 20
                                                                        PF1=Help

 1 Library       Specify source copybook libraries
 2 Record-type   Add/Delete record-types from COBOL/PL1 copybooks
 3 Replace       COBOL Replacing options
 4 Create        Create Structure (SDO) in the foreground
 5 Batch         Create Batch Job


Structure File to Create/Edit:            PDS/PDSE member
      Dsn> ______________________________________________
   Member> ________

_ Title      > ______________________________

_ Description> __________________________________________________________________
               __________________________________________________________________
```

Figure 107. SELCOPY/i - Create Structure from COBOL/PL1 copybook(s).

The Create Structure from COBOL/PL1 copybook(s) panel (ZZSGSDO1) is an interactive panel window, opened on selection of option 1. in the Create Structure (SDO) Menu.

Optional field entries must be activated by entering "/" in the preceding field if their values are to be included.

Options

1. Library
2. Record-type
3. Replace
4. Create
5. Batch

Create

Create the SDO structure file in the SELCOPY/i foreground using the current field values.

Batch

Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the SDE CREATE STRUCTURE command generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Panel Input Fields

Structure File to Create/Edit:

Fields that identify the SELCOPY/i structure (SDO) data set or PDS/PDSE member name. If the specified data set or PDS/PDSE member is an existing SDO which was generated **using this panel**, then panel options and field values will be re-populated using equivalent fields from the SDO.

Dsn>

Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be presented if wildcard character "*" occurs anywhere within the specified DSN.

Member>

If DSN is a PDS/PDSE library, specifies the SDO member name.

A selectable list of members will be presented if wildcard character "*" occurs anywhere within the specified member name or the member name is left blank.

This parameter field corresponds to CREATE STRUCTURE parameter *struct-name*.

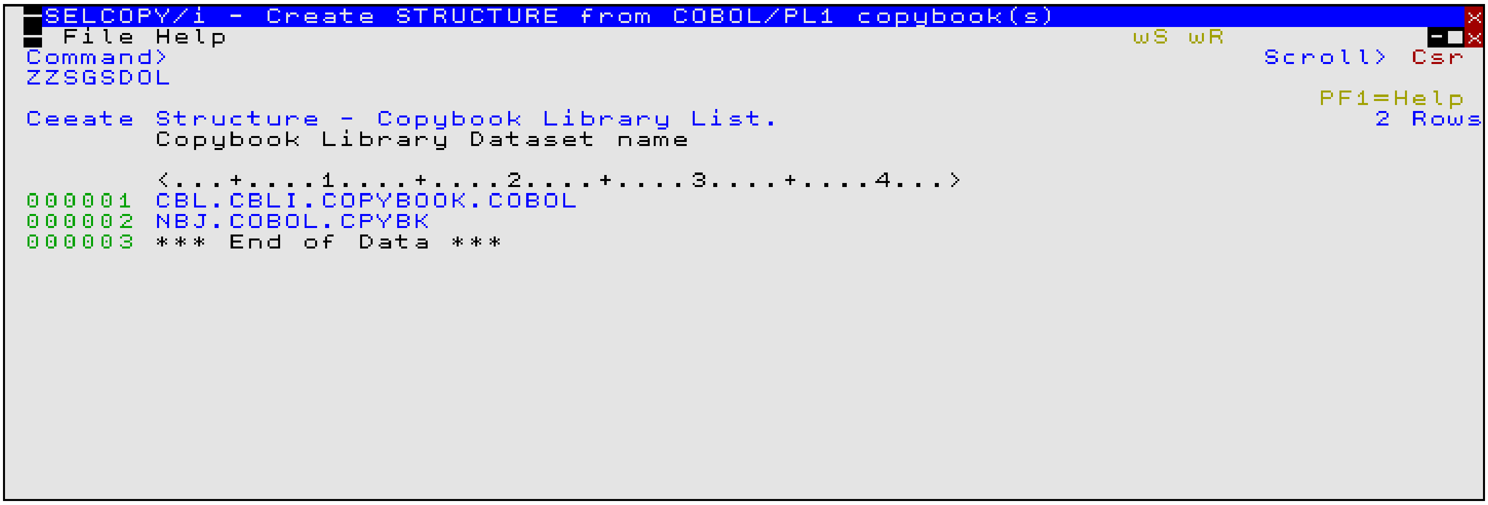
Title>

If activated, specifies an up to 30 character title for the SDO.
This parameter field corresponds to CREATE STRUCTURE parameter TITLE *sdo_title*.

Description>

If activated, specifies an up to 124 character description of the SDO.
This parameter field corresponds to CREATE STRUCTURE parameter DESCRIPTION *sdo_description*.

Create Structure - Copybook Library List



```
SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
 File Help                                                 wS wR
Command>                                                          Scroll> Csr
ZZSGSDOL
                                                                     PF1=Help
Ceeate Structure - Copybook Library List.                             2 Rows
       Copybook Library Dataset name

       <...+....1....+....2....+....3....+....4...>
000001 CBL.CBLI.COPYBOOK.COBOL
000002 NBJ.COBOL.CPYBK
000003 *** End of Data ***
```

Figure 108. SELCOPY/i - Create Structure - Copybook Library List.

The Create Structure - Copybook Library List panel (ZZSGSDOL) is an interactive panel window, opened on selection of option 1. in the Create Structure from COBOL/PL1 copybook(s) panel.

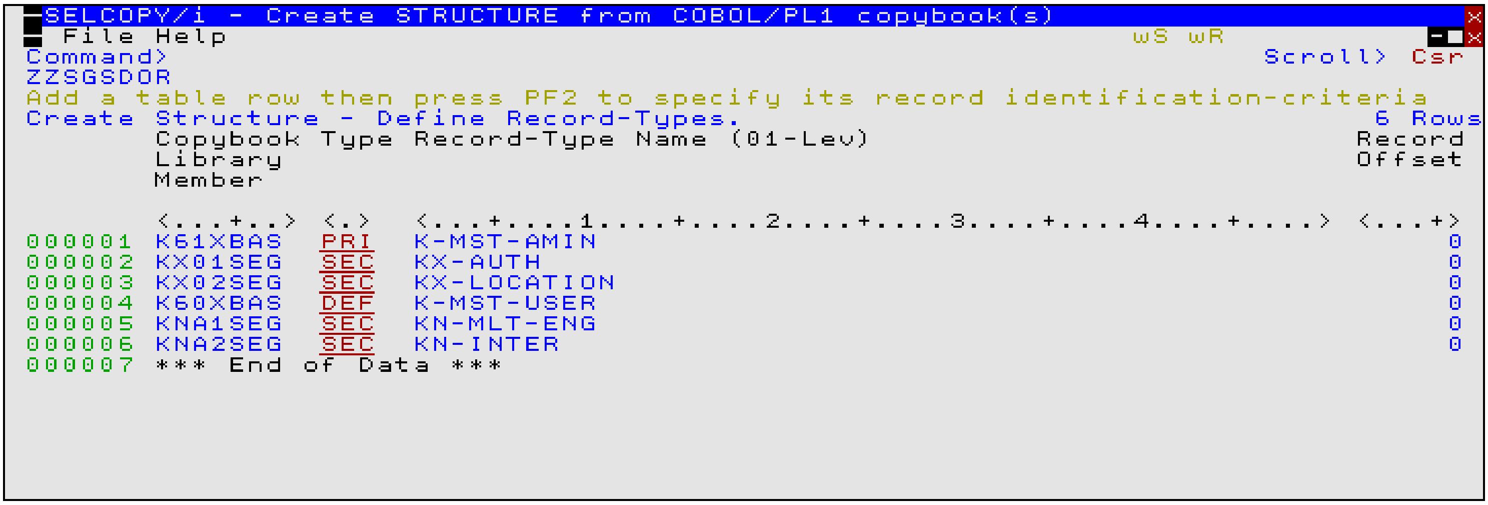
Standard SELCOPY/i table editing techniques should be used to add a table row entry for each required copybook source library DSN.

The table identifies library data sets and the order in which they are to be searched. The library search chain is used to locate each copybook source member specified in the Create Structure - Define Record Types panel.

Pressing <PF3> to exit the panel, will also save the table of libraries and return to the Create Structure from COBOL/PL1 copybook(s) panel.

These library names corresponds to CREATE STRUCTURE parameter LIBRARY(*search_lib* ...).

Create Structure - Define Record-Types



```
SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
 File Help                                                 wS wR
Command>                                                          Scroll> Csr
ZZSGSDOR
Add a table row then press PF2 to specify its record identification-criteria
Create Structure - Define Record-Types.                               6 Rows
       Copybook Type Record-Type Name (01-Lev)                      Record
       Library                                                      Offset
       Member

       <...+..> <.>  <...+....1....+....2....+....3....+....4....+....> <...+>
000001 K61XBAS  PRI  K-MST-AMIN                                             0
000002 KX01SEG  SEC  KX-AUTH                                                0
000003 KX02SEG  SEC  KX-LOCATION                                            0
000004 K60XBAS  DEF  K-MST-USER                                             0
000005 KNA1SEG  SEC  KN-MLT-ENG                                             0
000006 KNA2SEG  SEC  KN-INTER                                               0
000007 *** End of Data ***
```

Figure 109. SELCOPY/i - Create Structure - Define Record-Types.

The Create Structure - Define Record-Types panel (ZZSGSDOR) is an interactive panel window, opened on selection of option 2. in the Create Structure from COBOL/PL1 copybook(s) panel.

This panel is used to specify the source member and record type name for each record type definition (RTO) in the generated SDO.

Standard SELCOPY/i table editing techniques should be used to add a table row entry for each required record type definition.

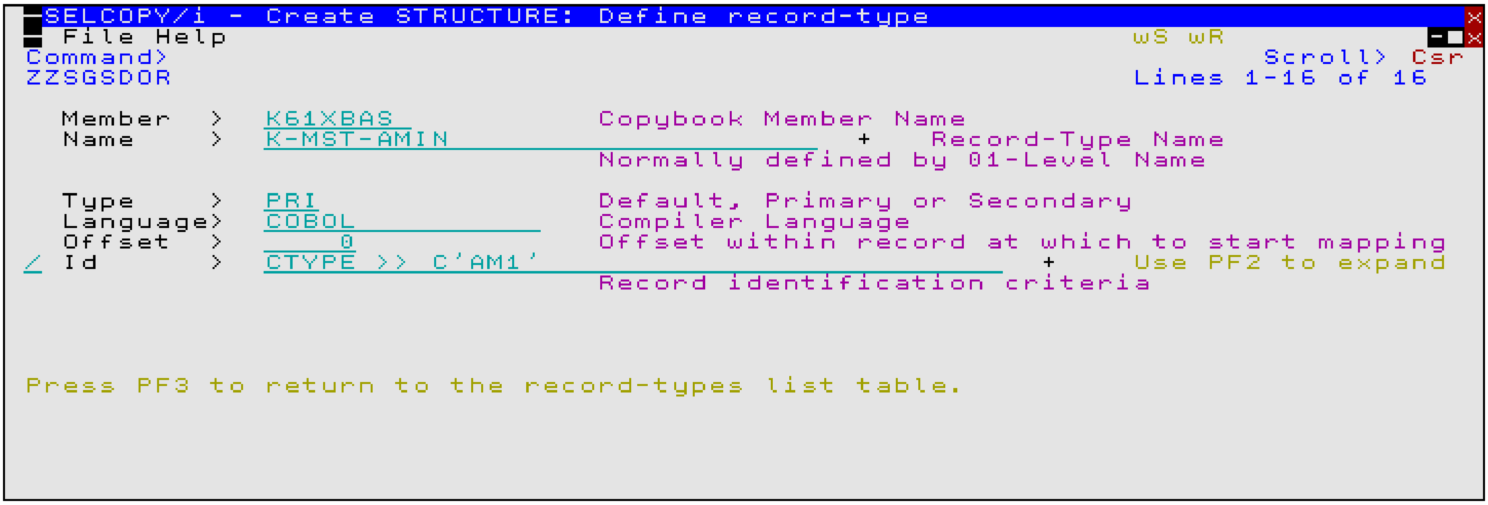
The specified record-type name must be the name of a COBOL GROUP field or PL1 STRUCTURE defined in the source copybook member. If the copybook member contains more than one group/structure field, each required to define a separate RTO, then the copybook member may be specified in more than one table row.

A record type definition must be defined as being either a primary segment (PRI), mapping an entire logical record or the first segment of a segmented logical record, or a secondary segment, mapping second and/or subsequent segments of a segmented logical record. At least one default (DEF) primary segment record type must also be defined.

Having inserted a table row, the user can display the single record view of the row using the ZOOM command (assigned to <PF2> by default.) The zoom view is required in order to identify the source member as being a PL1 copybook or ADATA output and also to to specify record type identification (USE WHEN) criteria.

Pressing <PF3> to exit the zoomed view of the panel, will update the record-type definition table row and return to the multi-record view of the table. Pressing <PF3> again to exit the Define Record-Types panel, will save the table of record type definitions and return to the Create Structure from COBOL/PL1 copybook(s) panel.

Fields in this table corresponds to CREATE STRUCTURE parameter RECORD (**Record Type Def**).



```
 SELCOPY/i - Create STRUCTURE: Define record-type
  File Help                                                 wS wR
Command>                                                           Scroll> Csr
ZZSGSDOR                                                    Lines 1-16 of 16

  Member   > K61XBAS          Copybook Member Name
  Name     > K-MST-AMIN                    +   Record-Type Name
                              Normally defined by 01-Level Name

  Type     > PRI              Default, Primary or Secondary
  Language> COBOL             Compiler Language
  Offset   >    0             Offset within record at which to start mapping
/ Id       > CTYPE >> C'AM1'                          +    Use PF2 to expand
                              Record identification criteria



Press PF3 to return to the record-types list table.
```

Figure 110. SELCOPY/i - Create Structure - Define Record-Types Zoomed View.

Panel Input Fields

Member>

Name of a COBOL or PL1, copybook or ADATA member belonging to a library in the defined library search chain. Note that, if the member name exists in more than one library in the search chain, then the first occurrence will be used.

This field corresponds to CREATE STRUCTURE parameter SOURCE *member_name*.

Name>

Name of the record type to be generated in the SDO. The specified name must match the name of a COBOL GROUP or PL1 STRUCTURE defined in the source member.

This field corresponds to CREATE STRUCTURE parameter NAME *record_type*.

Type>

Specifies the type (**DEF**, **PRI** or **SEC**) of the record-type to be defined.

PRI indicates that the record type maps an entire logical record or the first (primary) segment of a segmented logical record.
SEC indicates that the record type definition maps second and/or subsequent (secondary) segments of a segmented logical record.
DEF indicates that the record type is a default primary record type definition. One and only one DEF record type must be defined in the SDO.

This field corresponds to CREATE STRUCTURE parameters PRIMARY, SECONDARY and DEFAULT.

Language>

Specifies the format of the source mamber as a **COBOL** copybook, **PL1** include file or **ADATA** (COBOL or PL1) output file.

Note that it is possible to combine COBOL, PL1 and ADATA source members in the same SDO generation.

This field corresponds to CREATE STRUCTURE parameter SOURCE COBOL|PL1|ADATA.

Offset>

Optional positive (or negative) numeric value specifying the offset into (or before) the record/segment data at which the record type mapping will begin.

This field corresponds to CREATE STRUCTURE parameter OFFSET <+|-> *n_bytes*.

Id>

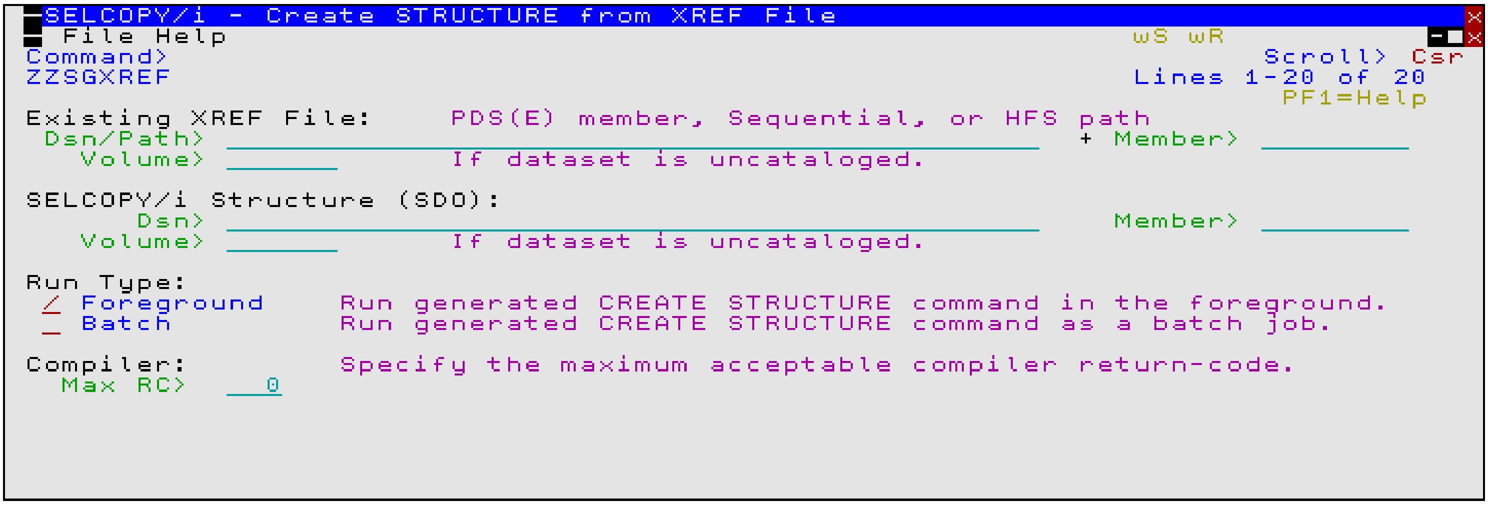
If activated, specifies arguments to a USE WHEN expression to be saved in the record type definition. The USE WHEN expression identifies the criteria, based on record data, for which this record type will be applied.

For non-segmented records or primary record segments (type DEF or PRI), the USE WHEN condition references data within the current record or segment only. However, a secondary segment (SEC) condition may also reference data in the primary segment or previous secondary segments.

Furthermore, for all segment types (DEF, PRI and SEC), the record data may be referenced using unformatted record positions or formatted field names. See information on SDE expressions with particular reference to field value terms and built-in functions.

This field corresponds to CREATE STRUCTURE parameter USE WHEN *expression*.

Create Structure from XREF File



```
 SELCOPY/i - Create STRUCTURE from XREF File
  File Help                                                    wS wR
Command>                                                              Scroll> Csr
ZZSGXREF                                                        Lines 1-20 of 20
                                                                         PF1=Help
Existing XREF File:     PDS(E) member, Sequential, or HFS path
 Dsn/Path> ____________________________________________  + Member> ________
   Volume> ______      If dataset is uncataloged.

SELCOPY/i Structure (SDO):
      Dsn> ____________________________________________    Member> ________
   Volume> ______      If dataset is uncataloged.

Run Type:
 / Foreground          Run generated CREATE STRUCTURE command in the foreground.
 _ Batch               Run generated CREATE STRUCTURE command as a batch job.

Compiler:              Specify the maximum acceptable compiler return-code.
  Max RC>  ___0
```

Figure 111. SELCOPY/i - Create Structure from XREF File.

The Create Structure from XREF File panel (ZZSGXREF) is an interactive panel window, opened on selection of option 2. in the Create Structure (SDO) Menu. This panel provides an interface to the ZZSXREF utility to convert XREF files to SELCOPY/i structure definition file (SDO).

Panel Input Fields

Existing XREF File:

Fields that identify the source XREF data set, HFS file or PDS/PDSE member.

Dsn/Path>

An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data set names or HFS files will be displayed as appropriate if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Member>

If DSN is a PDS/PDSE library, specifies the XREF member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

`Volume>`

Specifies a volume serial id mask for an uncataloged XREF file. (Not applicable to HFS files.)

`SELCOPY/i Structure (SDO):`

Fields that identify the SELCOPY/i structure (SDO) data set or PDS/PDSE member to be generated.

Beware that selecting an existing sequential data set or member will overwrite all existing data in that file.

`Dsn>`

Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be displayed if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified.

`Member>`

If DSN is an existing PDS/PDSE library, this field specifies the SDO member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

`Volume>`

Specifies a volume serial id mask for an uncataloged SDO.

`Run Type>`

Specifies whether the conversion utility is to be executed in the SELCOPY/i **Foreground** or will generate a JCL **Batch** job as a temporary file displayed in a CBLe text edit view.

The batch job runs SDEAMAIN to execute CREATE STRUCTURE syntax and may be submitted to batch using the SUBMIT command.

`Compiler:`

SELCOPY/i requires a COBOL or PL1 compilation ADATA output file in order to generate an SDO structure file. Therefore, the COBOL or PL1 compiler is invoked to compile all copybook files specified in the XREF file.

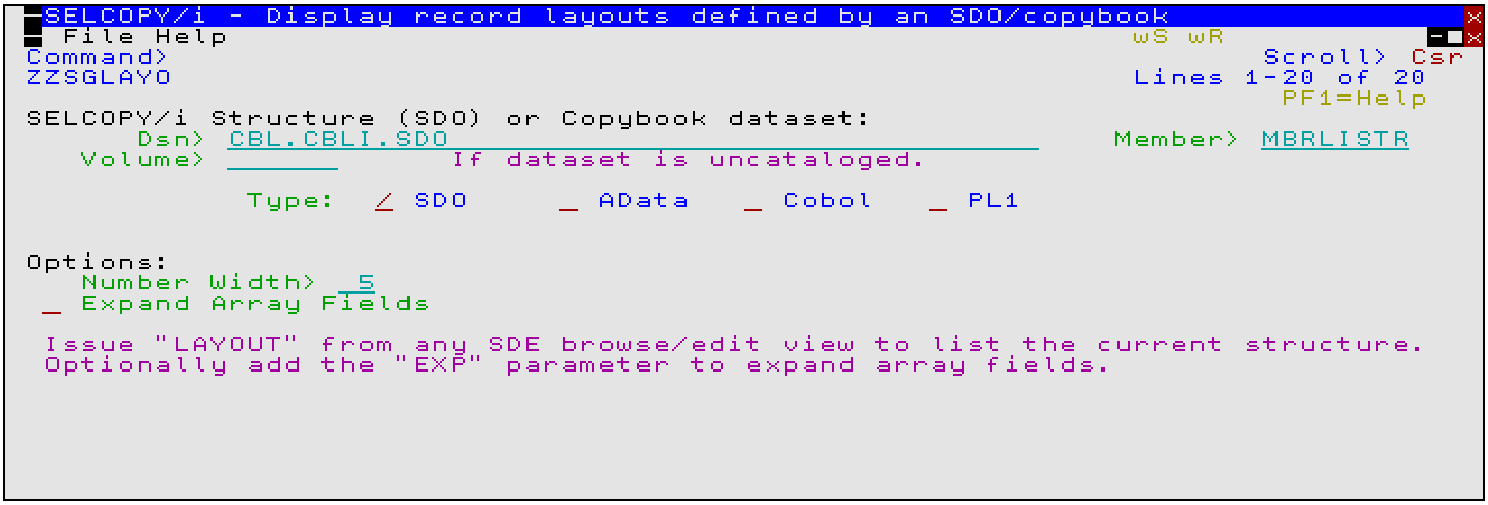
`Max RC>`

Specifies the maximum acceptable return code that may be returned by the COBOL or PL1 compilers in order for SELCOPY/i to continue generating the SDO from the resulting ADATA file.

Display Record Layout

Display Record Layout Panel

The Display Record Layout panel (ZZSGLAYO) is an interactive panel window, opened on selection of option 3. in the Create Structure (SDO) Menu. This panel provides an interface to the SDE LAYOUT command used to display the layout of all records within a record structure (SDO, COBOL/PL1 Copybook or COBOL/PL1 ADATA) file.



```
 SELCOPY/i - Display record layouts defined by an SDO/copybook
 File Help                                                   wS wR
Command>                                                        Scroll> Csr
ZZSGLAYO                                                    Lines 1-20 of 20
                                                                  PF1=Help
SELCOPY/i Structure (SDO) or Copybook dataset:
        Dsn> CBL.CBLI.SDO                                 Member> MBRLISTR
     Volume> ______          If dataset is uncataloged.

                Type: / SDO      _ AData    _ Cobol    _ PL1


Options:
   Number Width> _5
 _ Expand Array Fields

 Issue "LAYOUT" from any SDE browse/edit view to list the current structure.
 Optionally add the "EXP" parameter to expand array fields.
```

Figure 112. SELCOPY/i - Display Record Layout Panel.

Panel Input Fields

SELCOPY/i Structure (SDO) or Copybook dataset:

Fields that identify the source structure (SDO, Copybook or ADATA) data set or PDS/PDSE member that contain one or more record structure definitions.

Dsn>

Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be displayed if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Member>

If DSN is an existing PDS/PDSE library, this field specifies the member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>

Specifies a volume serial id mask for an uncataloged data set.

Type:

Specifies the format (SDO, **COBOL** or **PL1** Copybook, COBOL or PL1 **ADATA** output) of the record structure definition file.

Number Width>

Specifies the displayed width of numeric columns: **RefNo**, **Start**, **End** and **Length** in the layout list output.

The minimum width of these field is 5 characters and so this value need only be increased if it is known that the decimal display of these field values exceeds this width.

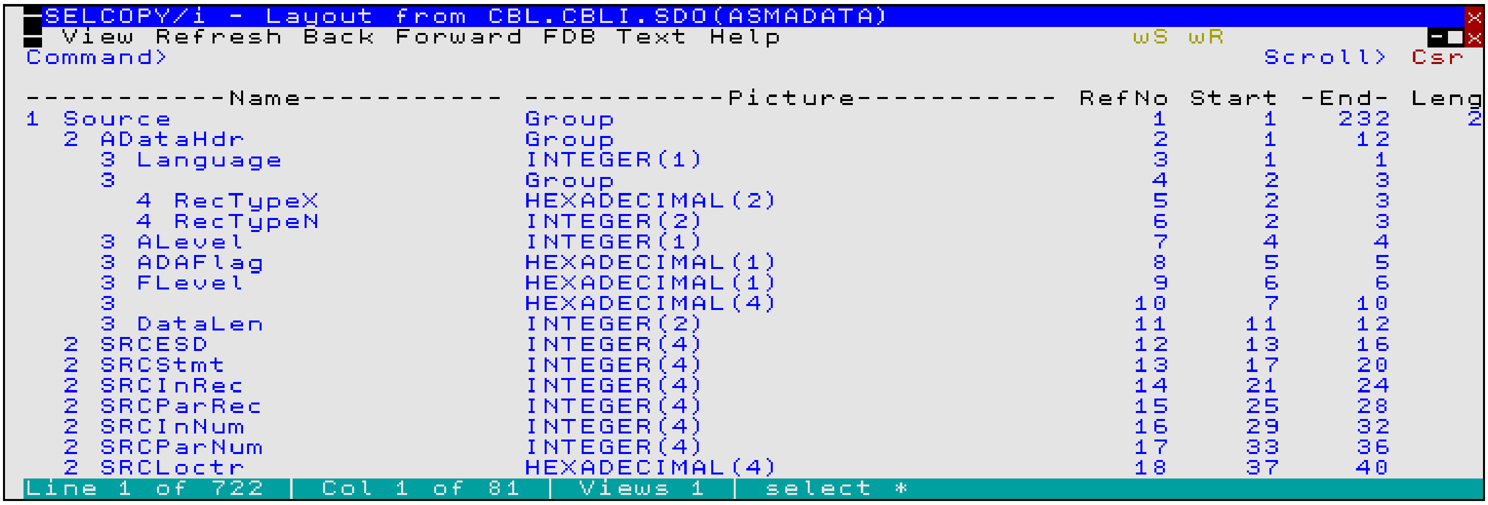
Expand Array Fields

This option field determines whether or not array (OCCURS) fields are expanded to display every repeating instance of a field within that array.

Display Record Layout Output

The structure layout is displayed in a list window.

For each field, the nested level number of that field is displayed to the left of the field name in the **Name** column. Furthermore, indentation occurs within the **Name** column for each nested level within a Group field.



```
 SELCOPY/i - Layout from CBL.CBLI.SDO(ASMADATA)
  View Refresh Back Forward FDB Text Help                   wS wR
Command>                                                          Scroll> Csr

-----------Name------------ -----------Picture----------- RefNo Start -End- Leng
1 Source                    Group                             1     1   232    2
  2 ADataHdr                Group                             2     1    12
    3 Language              INTEGER(1)                        3     1     1
    3                       Group                             4     2     3
      4 RecTypeX            HEXADECIMAL(2)                    5     2     3
      4 RecTypeN            INTEGER(2)                        6     2     3
    3 ALevel                INTEGER(1)                        7     4     4
    3 ADAFlag               HEXADECIMAL(1)                    8     5     5
    3 FLevel                HEXADECIMAL(1)                    9     6     6
    3                       HEXADECIMAL(4)                   10     7    10
    3 DataLen               INTEGER(2)                       11    11    12
  2 SRCESD                  INTEGER(4)                       12    13    16
  2 SRCStmt                 INTEGER(4)                       13    17    20
  2 SRCInRec                INTEGER(4)                       14    21    24
  2 SRCParRec               INTEGER(4)                       15    25    28
  2 SRCInNum                INTEGER(4)                       16    29    32
  2 SRCParNum               INTEGER(4)                       17    33    36
  2 SRCLoctr                HEXADECIMAL(4)                   18    37    40
Line 1 of 722 | Col 1 of 81 | Views 1 | select *
```

Figure 113. SELCOPY/i - Display Record Layout Output.

Columns Displayed

Name	Type	Description
Name	ALPair	Field level and name
Picture	ALPair	Field picture and data type
RefNo	Int	Field reference number
Start	Int	Field start position
End	Int	Field end position
Length	Int	Field length

Create File Filter

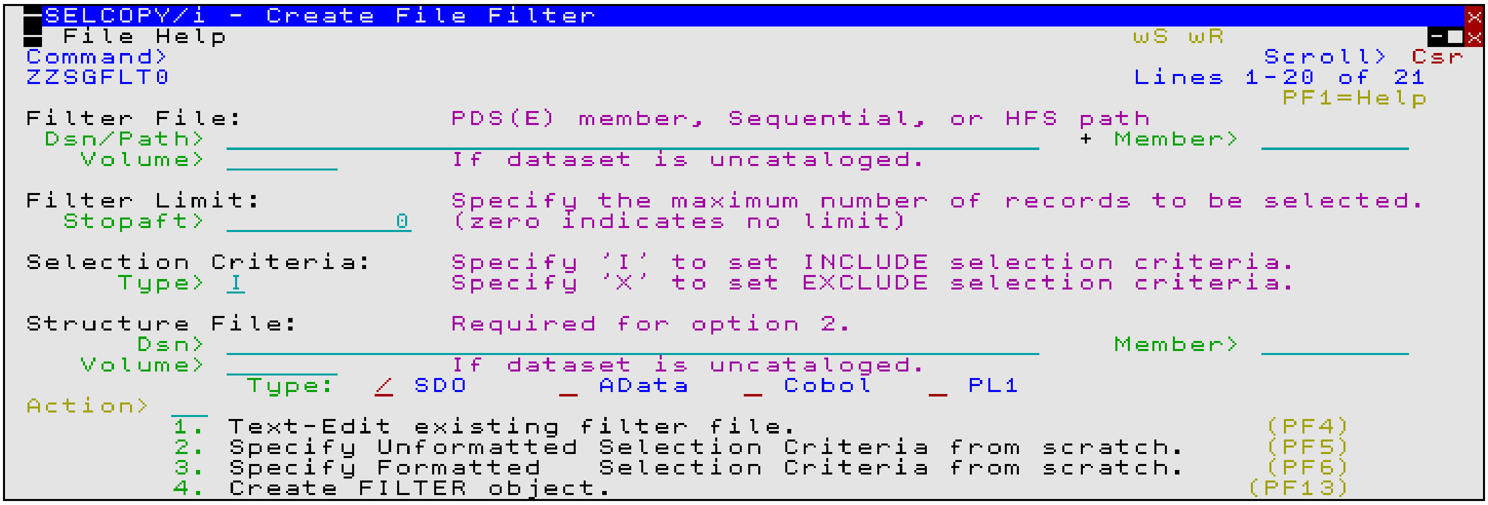
SDE structured data BROWSE and update in-place EDIT supports use of a filter so that only records that satisfy criteria defined by the filter clause are eligible for display in the SDE window view.

A filter clause may be passed to the SDE EDIT or BROWSE operation via in-line command syntax or via a filter file. A filter file contains the filter syntax that would be specified in-line but has the advantage that it may be kept for use in other EDIT and BROWSE operations.

The Create File Filter panel assists with generation of a filter clause which is ultimately saved to a filter file. The filter file may then be specified on the FILTER: field of the **SDE - Structured Data Browse/Edit** panel.

The Create File Filter Dialog panel (ZZSGFLT0) is an interactive panel window and may be started via the following:

- Select option 10. in the SELCOPY/i Primary option menu.
- Select 'Create File Filter' from the Utilities menu.
- Select menu item **FILTER** from the SDE - Structured Data Browse/Edit panel.
- Execute the command **FILTER** with no parameters from the command line of any window.



```
 SELCOPY/i - Create File Filter
  File Help                                                      wS wR
Command>                                                               Scroll> Csr
ZZSGFLT0                                                          Lines 1-20 of 21
                                                                          PF1=Help
Filter File:              PDS(E) member, Sequential, or HFS path
  Dsn/Path> ____________________________________________  + Member> ________
    Volume> ______        If dataset is uncataloged.

Filter Limit:             Specify the maximum number of records to be selected.
   Stopaft> _________0    (zero indicates no limit)

Selection Criteria:       Specify 'I' to set INCLUDE selection criteria.
        Type> I           Specify 'X' to set EXCLUDE selection criteria.

Structure File:           Required for option 2.
        Dsn> ____________________________________________     Member> ________
    Volume> ______        If dataset is uncataloged.
          Type:   / SDO      _ AData    _ Cobol    _ PL1
Action>
        __
        1. Text-Edit existing filter file.                               (PF4)
        2. Specify Unformatted Selection Criteria from scratch.          (PF5)
        3. Specify Formatted   Selection Criteria from scratch.          (PF6)
        4. Create FILTER object.                                        (PF13)
```

Figure 114. SELCOPY/i - Create File Filter.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time the panel was used.

Filter File:

Fields that together identify the PDS/PDSE library member, sequential data set or HFS file path to be edited, using the CBLe text editor, and in which a generated filter clause will be copied (replacing any existing data).

This file is not saved to disk before it is displayed, however, the user will be prompted to save it on exit (PF3).

Dsn/Path>

An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data set names or HFS files will be displayed as appropriate if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Member>

If DSN is a PDS/PDSE library, specifies the FILTER file member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>

Specifies a volume serial id mask for an uncataloged FILTER file. (Not applicable to HFS files.)

Filter Limit:

The filter limit defines when to stop filtering input records.

Stopaft>

Specifies the maximum number of records to be selected by this filter. If this threshold is reached, then:

- For an INCLUDE filter, all remaining untested records are excluded.
- For an EXCLUDE filter, all remaining untested records are included.

A value of 0 (zero) removes this filter limit.

Selection Criteria:

Determines whether filtering is to include or exclude records that match the filter clause record selection criteria.

Type>

Specifies either **I** (INCLUDE) or **X** (EXCLUDE) to indicate the action performed on records that selected by the filter clause.

Structure File:

Fields that identify the sequential data set name or PDS/PDSE member name of a file containing record structure definitions to be used to select formatted field names in the filter clause WHERE expression. (See Action 3. "*Specify Formatted Selection Criteria*".)

Dsn>

Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be displayed if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified.

Member>

If DSN is an existing PDS/PDSE library, this field specifies the member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>

Specifies a volume serial id mask for an uncataloged data set.

Type:

Specifies the format (SDO, **COBOL** or **PL1** Copybook, COBOL or PL1 **ADATA** output) of the record structure definition file.

Action>

Option field specifying the action to be taken when the <Enter> key is pressed.

1. **Text-Edit existing filter file.**
   Opens a CBLe text edit view to display and allow edit of a filter clause specified in an existing file identified by Filter File.

   No new filter is generated as a result of selecting action 1. so the filter expresions that already exist in the file are preserved.

2. **Specify Unformatted Selection Criteria.**
   Opens the Filter (unformatted) - Selection Criteria panel to build filter clause record selection criteria based on unformatted record data.

3. **Specify Formatted Selection Criteria.**
   Opens the Filter (formatted) - Selection Criteria panel to build filter clause record selection criteria based on record data which has been formatted using a Structure File.

   This panel will first display a table view of all record structures defined within the structure file. Select the record structures that are to be included or excluded.

   For each of the selected record structures, further record selection criteria may then be specified based on fields defined within the record structures.

4. **Create FILTER object.**
   Opens a CBLe text edit view to display the generated filter clause in the file identified by Filter File.

   When the edit window is closed (<PF3>), the user will be prompted to save the file and, if necessary, open the Allocate NonVSAM dialog window.

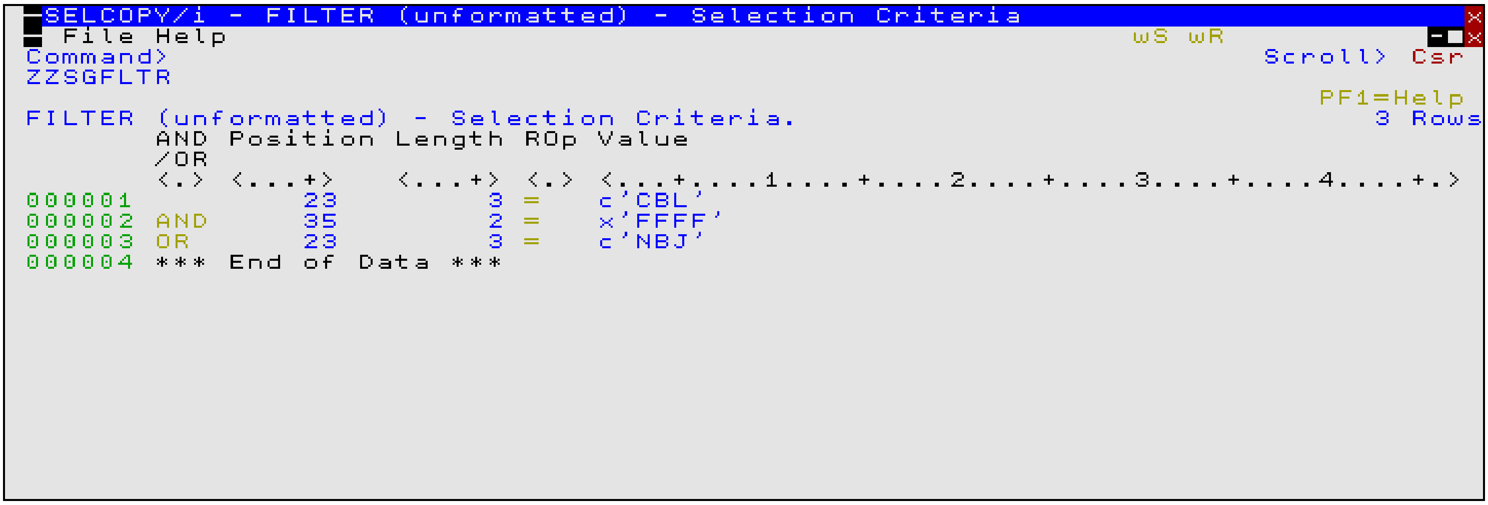
   In order to prevent accidental overwrite of data, if the file already exists, then the save will be rejected with message ZZSE046E. To force the save regardless of the warning, issue command **SSave,** (or **FFile** to save and exit at once).

Unformatted Selection Criteria

The FILTER (unformatted) - Selection Criteria panel (ZZSGFLTR) is an interactive panel window, opened on selecting Action 2. in the Create File Filter panel.

The panel contains a table where each row represents a sub-expression of the single INCLUDE or EXCLUDE sub-clause (WHERE expression). Each generated sub-expression involves a SUBSTR() built-in function which operates on the unformatted record data referenced by "record".

The sub-expressions are separated by logical operator AND or OR, where AND is higher in the order of operator precedence than OR. i.e. A AND B OR C is equivalent to (A AND B) OR C.



```
SELCOPY/i - FILTER (unformatted) - Selection Criteria
 File Help                                                   wS wR
Command>                                                            Scroll> Csr
ZZSGFLTR
                                                                       PF1=Help
FILTER (unformatted) - Selection Criteria.                               3 Rows
       AND Position Length ROp Value
       /OR
       <.>  <...+>    <...+> <.> <...+....1....+....2....+....3....+....4....+.>
000001          23         3 =    c'CBL'
000002 AND      35         2 =    x'FFFF'
000003 OR       23         3 =    c'NBJ'
000004 *** End of Data ***
```

Figure 115. SELCOPY/i - Unformatted Selection Criteria.

Standard SELCOPY/i table editing techniques should be used to add a table row entry for each filter sub-expression.

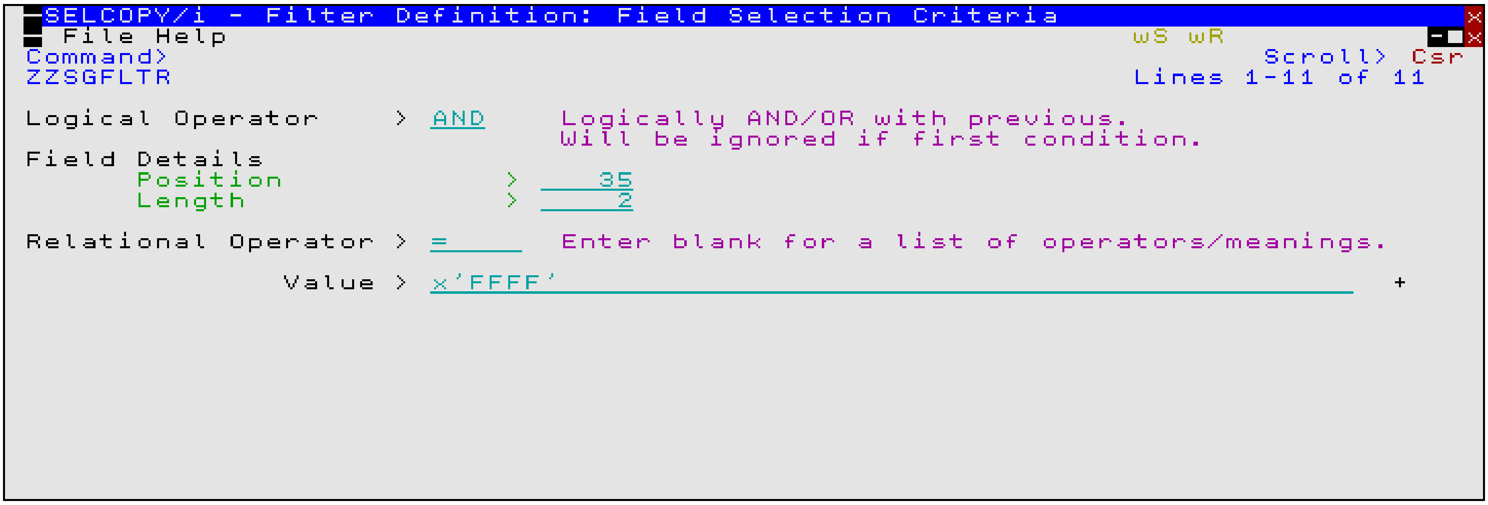
Each table row consists of the logical operator (AND/OR) position and length of the data within the record to be tested, the relational operator (ROp) and the value to be tested. Enter null or invalid enties in the **AND**/**OR** and **ROp** fields to display and select permitted entries for these fields.

Since logical operators AND and OR are dyadic, the **AND**/**OR** field of the first table row (the first term of the expression) is always blank and cannot be updated.

Having inserted a table row, the user can display the single record view of the row using the ZOOM command (assigned to <PF2> by default.)

Pressing <PF3> to exit the zoomed view of the panel, will update the filter sub-expression table row and return to the multi-record view of the table. Pressing <PF3> again to exit the Selection Criteria panel, will save the table sub-expressions and return to the Create File Filter panel.

Fields in this table corresponds to the filter WHERE *expression*.



```
SELCOPY/i - Filter Definition: Field Selection Criteria
 File Help                                                   wS wR
Command>                                                            Scroll> Csr
ZZSGFLTR                                                         Lines 1-11 of 11

Logical Operator     > AND     Logically AND/OR with previous.
                               Will be ignored if first condition.
Field Details
          Position         >      35
          Length           >       2

Relational Operator > =        Enter blank for a list of operators/meanings.

                 Value > x'FFFF'                                                 +
```

Figure 116. SELCOPY/i - Unformatted Selection Criteria Zoomed View.

Panel Input Fields

By default, field entries are populated with arguments and options entered in the table view for that entry.

Logical Operator>
Logical operator that identifies the relationship of this field test sub-expression with the previously specified sub-expression. If this is the first, then this field is ignored. Permissible operators are AND or OR.

Field Details:
Fields that together identify the location within the input records of the data field to be tested.

Position>
Integer numeric identifying the position of the first byte of the test field within the record data.

Length>
Integer numeric identifying the length of the test field within the record data.

Relational Operator>
Identifies the type of test to be performed on the field data. Enter blank in this field to display a selectable list of supported relational operator symbols and a brief description of each.

Value>
Specifies the literal string term against which the field data will be tested.

Formatted Selection Criteria

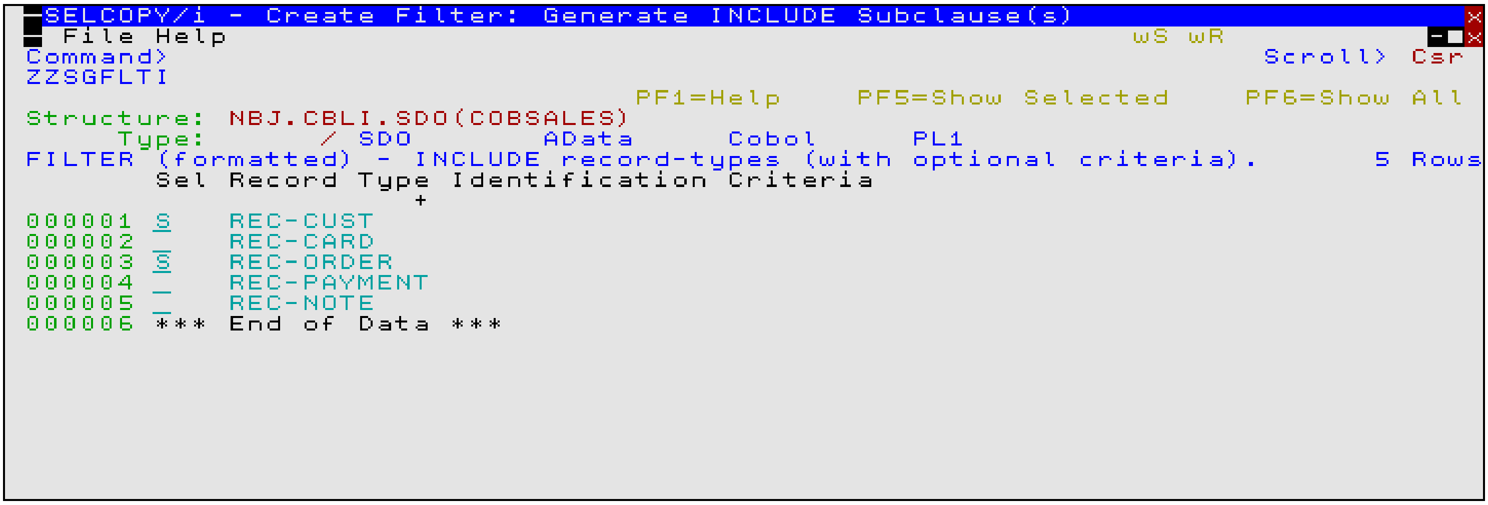
The Formatted Selection Criteria panels relate specifically to generating record selection criteria based on assigned record structure and, optionally, formatted field data.

FILTER (formatted) - INCLUDE/EXCLUDE record-types

The FILTER (formatted) - INCLUDE/EXCLUDE record-types panel (ZZSGFLTI) is an interactive panel window, opened on selecting Action 3. in the Create File Filter panel.

The type of filter (INCLUDE or EXCLUDE) is determined by the selection criteria type specified in the **Create File Filter** panel.

The panel contains a table where each selected and non-excluded row represents an INCLUDE or EXCLUDE sub-clause based on a record structure defined within the structure file. By default, the table contains one row for each record structure defined within the structure file.



```
 SELCOPY/i - Create Filter: Generate INCLUDE Subclause(s)
  File Help                                                 wS wR
Command>                                                            Scroll> Csr
ZZSGFLTI
                                     PF1=Help    PF5=Show Selected    PF6=Show All
Structure: NBJ.CBLI.SDO(COBSALES)
     Type:         / SDO       AData     Cobol     PL1
FILTER (formatted) - INCLUDE record-types (with optional criteria).       5 Rows
       Sel Record Type Identification Criteria
                      +
000001 S   REC-CUST
000002 _   REC-CARD
000003 S   REC-ORDER
000004 _   REC-PAYMENT
000005 _   REC-NOTE
000006 *** End of Data ***
```

Figure 117. SELCOPY/i - Formatted - INCLUDE/EXCLUDE record-types.

Standard SELCOPY/i table editing techniques should be used to duplicate, move or exclude table row entries for each INCLUDE/EXCLUDE sub-clause.

Select a record structure table entry for INCLUDE/EXCLUDE by entering "S" in the "Sel" column or placing the cursor on the table entry and pressing the <Enter> key or, if configured, double-clicking the left mouse button. To de-select a record structure table entry, simply remove the "S" from the "Sel" column.

<PF5> and <PF6> may be used to alternate between display of only selected rows and all rows in the table respectively. This has no effect on the generated filter, but is useful as a visual aid.

"Record Type" identifies the record structure name and "Identification Criteria" displays any USE WHEN record type criteria used to identify when the record structure is assigned to a record. These are output fields may not be updated. "Selection Criteria" indicates the number of potential field name based sub-expressions defined for the record structure.

On selecting a record structure, the Filter (formatted) - Selection Criteria is opened automatically, giving the user the option to further specify selection criteria based on named fields within the formatted record.

When the filter is generated, each INCLUDE/EXCLUDE sub-clause is separated by a logical OR operation. Therefore, if required, a record structure table row entry may be duplicated in order to provide alternative INCLUDE/EXCLUDE sub-clause record selection criteria based on different values within fields belonging to the same record structure.

Pressing <PF3> to exit the panel, will save the table INCLUDE/EXCLUDE sub-clauses and return to the Create File Filter panel.

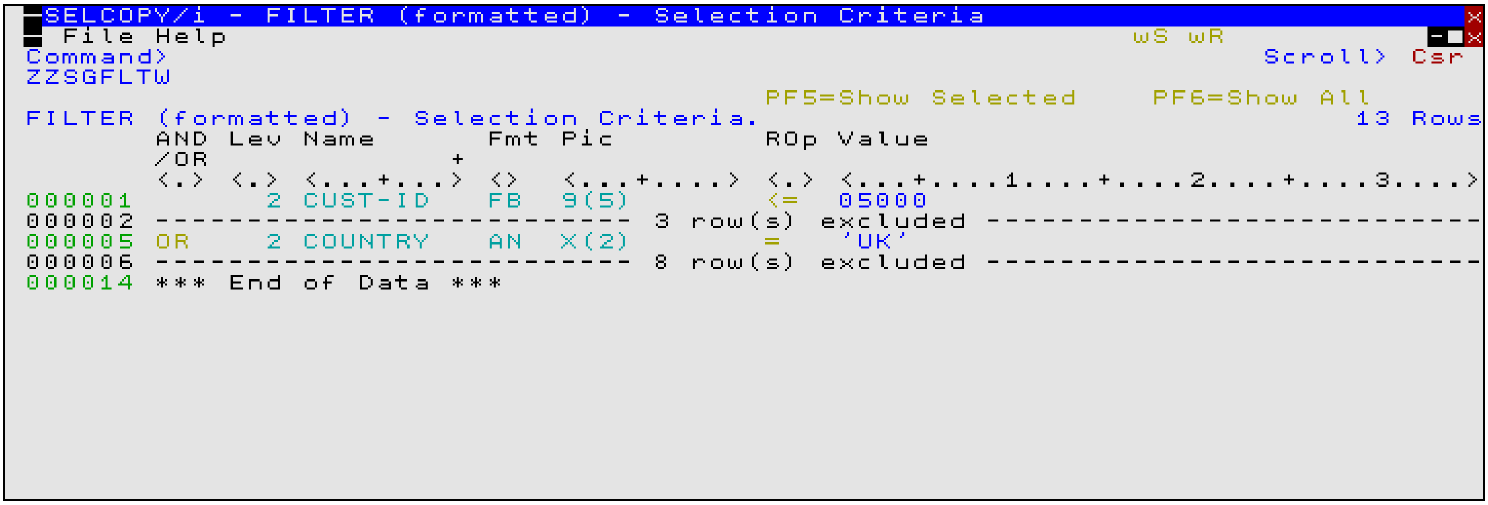
Filter (formatted) - Selection Criteria

The FILTER (formatted) - Selection Criteria panel (ZZSGFLTW) is an interactive panel window, opened when a record structure is selected from the FILTER (formatted) - INCLUDE/EXCLUDE record-types panel.

The panel contains a table where each row represents a sub-expression of the single INCLUDE or EXCLUDE sub-clause (WHERE expression) generated for the selected record structure. Each generated sub-expression is based on values referenced by one or more field names defined in the record structure.

By default, the table contains one row for every level of nested field defined within the record structure. These entries may be duplicated, copied or re-ordered before generating the sub-expression. Note that table entries that do not have a test value will **not** be included as part of the generated sub-expression.

The sub-expressions are separated by logical operator AND or OR, where AND is higher in the order of operator precedence than OR. i.e. A AND B OR C is equivalent to (A AND B) OR C. To filter on fields A AND (B OR C), then two rows should exist for the same record structure in the **FILTER (formatted) - INCLUDE/EXCLUDE record-types** table so that one tests fields A AND B and the other tests fields A AND C.



```
 SELCOPY/i - FILTER (formatted) - Selection Criteria
  File Help                                                 wS wR
Command>                                                           Scroll>  Csr
ZZSGFLTW
                                        PF5=Show Selected    PF6=Show All
FILTER (formatted) - Selection Criteria.                                13 Rows
       AND Lev Name      Fmt Pic        ROp Value
       /OR             +
       <.> <.> <...+...> <>  <...+....> <.> <...+....1....+....2....+....3....>
000001       2 CUST-ID   FB  9(5)       <=  05000
000002 -------------------------- 3 row(s) excluded ----------------------------
000005 OR    2 COUNTRY   AN  X(2)       =   'UK'
000006 -------------------------- 8 row(s) excluded ----------------------------
000014 *** End of Data ***
```

Figure 118. SELCOPY/i - Formatted Selection Criteria.

Standard SELCOPY/i table editing techniques should be used to update, move, copy and exclude table row entries for each filter sub-expression.

Each table row contains input fields for logical operator (AND/OR), relational operator (ROp) and the value against which the field will be tested. Null or invalid enties may be entered in the **AND**/**OR** and **ROp** fields to display and select permitted entries for these fields.

All other fields are output (non-updatable) fields providing useful information about the field to be tested (i.e. the field's level of nesting, its name, data type (format) and, if defined, its picture definition.)

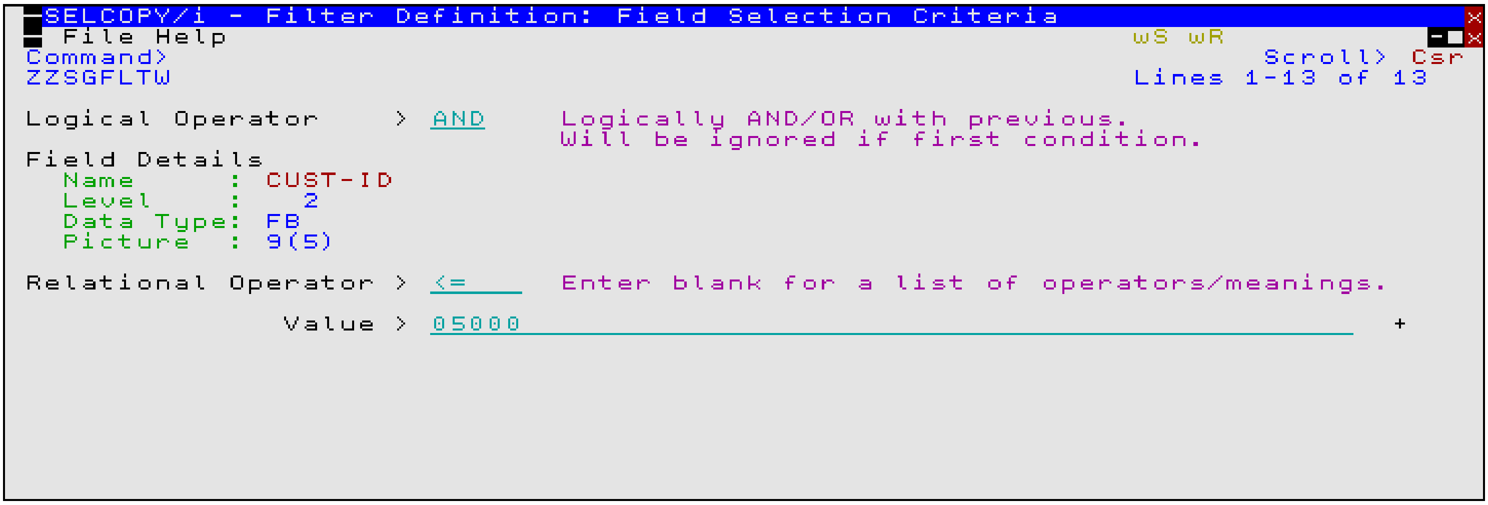
Since logical operators AND and OR are dyadic, the **AND**/**OR** field of the first table row (the first term of the expression) is always blank and cannot be updated.

<PF5> and <PF6> may be used to alternate between display of only selected rows and all rows in the table respectively. This has no effect on the generated filter, but is useful as a visual aid.

The user can display the single record view of the row using the ZOOM command (assigned to <PF2> by default.) As well providing additional helpful comment information, the single record view allows the user to use the panel EXPAND feature to enter a Value field entry that is longer than the provided input area.

Pressing <PF3> to exit the zoomed view of the panel, will update the filter sub-expression table row and return to the multi-record view of the table. Pressing <PF3> again to exit the Selection Criteria panel, will save the table sub-expressions and return to the FILTER (formatted) - INCLUDE/EXCLUDE record-types panel.

Fields in this table corresponds to the filter WHERE *expression*.



```
 SELCOPY/i - Filter Definition: Field Selection Criteria
  File Help                                                    wS wR
Command>                                                              Scroll> Csr
ZZSGFLTW                                                         Lines 1-13 of 13

Logical Operator     > AND     Logically AND/OR with previous.
                               Will be ignored if first condition.
Field Details
   Name      :  CUST-ID
   Level     :    2
   Data Type:  FB
   Picture   :  9(5)

Relational Operator > <=       Enter blank for a list of operators/meanings.

              Value > 05000                                                  +
```

Figure 119. SELCOPY/i - Formatted Selection Criteria Zoomed View.

Panel Input Fields

By default, field entries are populated with arguments and options entered in the table view for that entry.

`Logical Operator>`

Logical operator that identifies the relationship of this field test sub-expression with the previously specified sub-expression. If this is the first, then this field is ignored. Permissible operators are AND or OR.

`Field Details:`

Informational output fields that describe the formatted field.

`Name:`

Field name.

`Level:`

Level of nesting below the first level (GROUP or STRUCTURE) field.

`Data Type:`

The field's defined data type. See *"SDE Data Types"* for details of supported data types and their abbreviated names.

`Picture:`

If the source field is defined with a COBOL picture string, its representation is displayed in this field.

`Relational Operator>`

Identifies the type of test to be performed on the field data. Enter blank in this field to display a selectable list of supported relational operator symbols and a brief description of each.

`Value>`

Specifies the literal string term against which the field data will be tested.

Create New Datasets Menu (=11)

The Create New Datasets Menu panel (ZZSGDEFN) is an interactive panel window opened on selection of option 11. in the SELCOPY/i Primary option menu.

New files may be defined to the system from within the SELCOPY/i environment.

Note that "Copy" automatically invokes an Allocate Non-VSAM or Deine VSAM object panel to create a new output data set if required.

Options

1 Non-VSAM	ALLOC - Allocate new Sequential or PDS/PDSE library
2 KSDS	AMSK - Define new VSAM KSDS
3 ESDS	AMSE - Define new VSAM ESDS
4 RRDS	AMSR - Define new VSAM RRDS
5 LDS	AMSL - Define new VSAM LDS
6 ALIAS	AMSA - Defines new Catalog Alias
7 Copy	FC - Copy an existing dataset

Allocate NonVSAM (=11.1)

The Allocate NonVSAM Dialog window may be opened via the following:

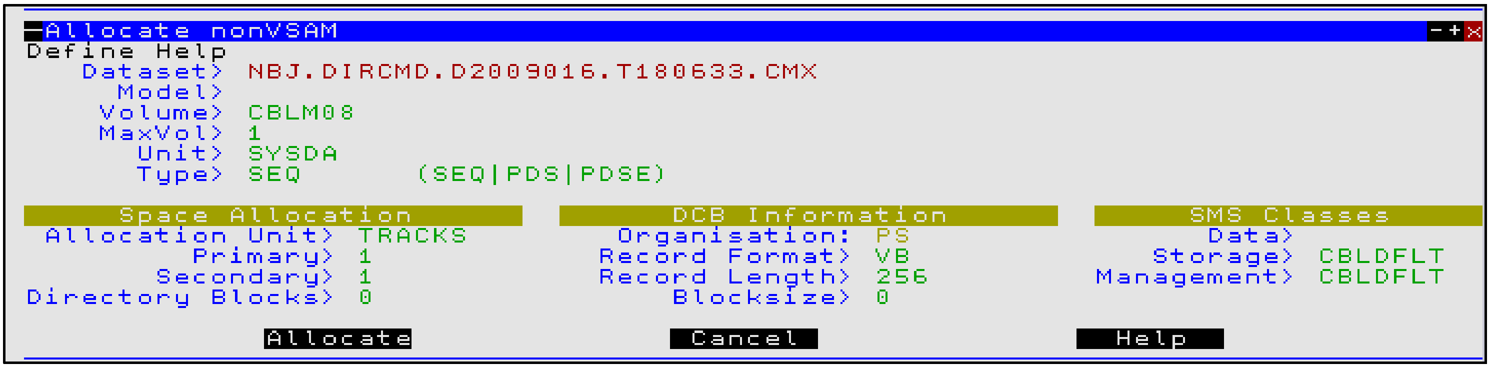
- Select 'Non-VSAM' from the Create New Datasets menu panel.
- Select 'Allocate NonVSAM' from the File menu in the CBLe main window menu bar.
- By any SELCOPY/i utility that requires allocation of an output data set.
- Enter the CBLe command ALLOCATE with no parameters on the command line of any window.
- Perform CBLe or SDE Edit using a new data set name, add some data and SAVE.
- Perform CBLe or SDE Edit of an existing non-VSAM data set, execute SET FILEID to assign a new DSN to the data in storage then SAVE.
  Note that the original data set is unchanged.

The Allocate nonVSAM window allows the user to supply characteristics for a new cataloged non-VSAM data set, then select the Allocate button or the Define menu item to action the allocation.

Fields within these dialogs represent the relevant TSO ALLOCATE or JCL DD statement parameters as appropriate for a new cataloged non-VSAM data set. Please refer to the *"TSO/E Command Reference"* and *"MVS JCL Reference"* for further information.

The **Model>** field allows the user to model the new entry's characteristics on an existing cataloged data set entry. On entering a non-VSAM data set name in the Model field and hitting <Enter>, all other fields are updated automatically to reflect the inherited values.

Note: Not implemented for CMS and VSE.



```
=Allocate nonVSAM                                                    -+x
Define Help
   Dataset> NBJ.DIRCMD.D2009016.T180633.CMX
     Model>
    Volume> CBLM08
    MaxVol> 1
      Unit> SYSDA
      Type> SEQ        (SEQ|PDS|PDSE)

    Space Allocation             DCB Information               SMS Classes
 Allocation Unit> TRACKS       Organisation: PS                   Data>
        Primary> 1            Record Format> VB               Storage> CBLDFLT
      Secondary> 1            Record Length> 256           Management> CBLDFLT
Directory Blocks> 0               Blocksize> 0

             Allocate                    Cancel                    Help
```

Figure 120. Allocate nonVSAM window.

Menu Bar Items

`Define`

Drop down menu containing the following items:

`Foreground`

Applicable to operation under TSO and CMS. Allocate the data set in the foreground (Control is temporarily passed to TSO or CMS).

Foreground+IEBCOPY (Not yet enabled)
As for Foreground but also copy data from the data set specified in the Model field.

Background (Not yet enabled)
Applicable to operation on VSE and MVS. A CBLe view is opened to edit a temporary job containing batch JCL to allocate/define the data set. The job may be submitted to the batch system using the CBLe command SUB.

Background+IEBCOPY (Not yet enabled)
As for Background but also include JCL to populate the data set with data from the data set specified in the Model field.

Help
Open the help window for data set allocation.

Define VSAM KSDS/ESDS/RRDS/LDS (=11.2/3/4/5)

The Define VSAM KSDS/ESDS/RRDS/LDS Dialog windows may be opened via the following:

- Select 'KSDS', 'ESDS', 'RRDS' or 'LDS' as appropriate from the Create New Datasets menu panel.
- Select 'Define KSDS', 'Define ESDS', 'Define RRDS' or 'Define LDS' as appropriate from the File menu in the CBLe main window menu bar.
- Enter line command AMSDIALOG with option KSDS, ESDS, RRDS or LDS on the command line of any window. Alternatively, use the synonyms AMSK, AMSE, AMSR or AMSL respectively.
- Perform CBLe or SDE Edit using a new data set name, execute SET DSORG KSDS/ESDS/RRDS/LDS, as required, add some data and SAVE.
- Perform CBLe or SDE Edit of an existing VSAM data set, execute SET FILEID to assign a new DSN to the data in storage then SAVE to save the data to a new VSAM data set of the same type.
  Note that the original VSAM data set is unchanged.

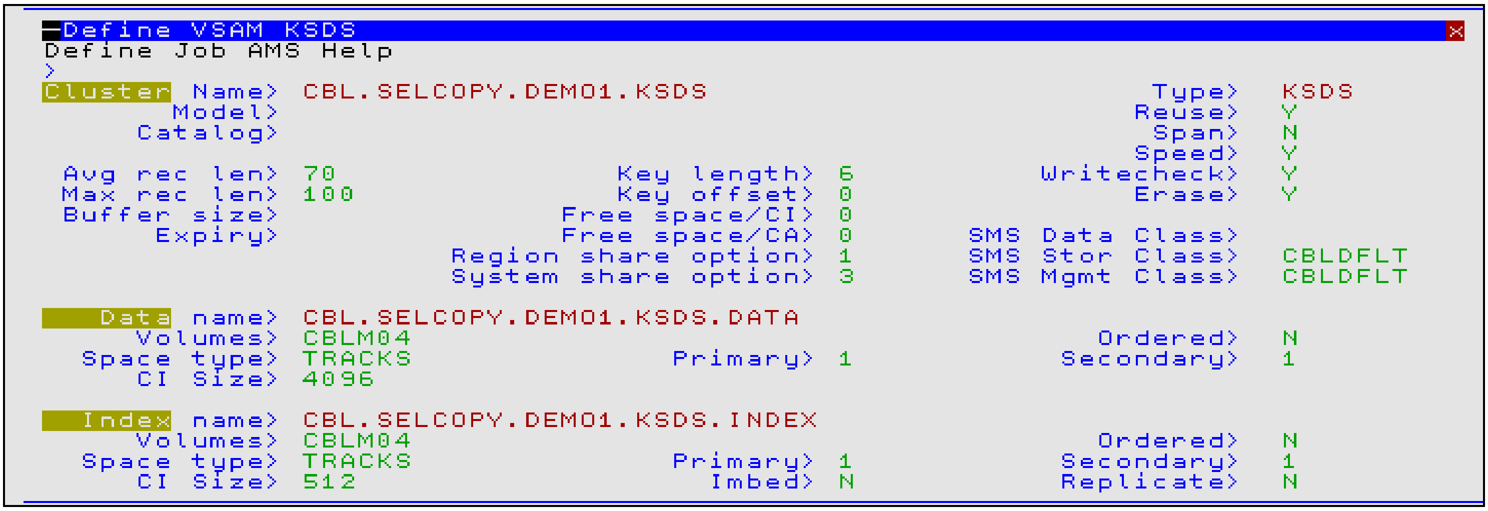
Define VSAM dialog windows allow the user to supply IDCAMS DEFINE characteristics for a new VSAM CLUSTER.

Select the appropriate menu bar item (see below) to define the new entry.

Fields within these dialogs represent the relevant IDCAMS DEFINE CLUSTER parameters as appropriate for the entry being defined. Please refer to *"DFSMS Access Method Services for Catalogs"* for further information.

The **Model>** field allows the user to model the new entry's characteristics on an existing catalog entry. On entering a VSAM data set name in the Model field and hitting <Enter>, all other fields are updated automatically to reflect the inherited values.

Note: Not implemented for CMS and VSE.



```
 Define VSAM KSDS
Define Job AMS Help
>
Cluster Name> CBL.SELCOPY.DEMO1.KSDS                        Type> KSDS
       Model>                                              Reuse> Y
     Catalog>                                               Span> N
                                                           Speed> Y
 Avg rec len> 70               Key length> 6          Writecheck> Y
 Max rec len> 100              Key offset> 0               Erase> Y
 Buffer size>               Free space/CI> 0
      Expiry>               Free space/CA> 0     SMS Data Class>
                      Region share option> 1     SMS Stor Class> CBLDFLT
                      System share option> 3     SMS Mgmt Class> CBLDFLT

   Data name> CBL.SELCOPY.DEMO1.KSDS.DATA
     Volumes> CBLM04                                     Ordered> N
  Space type> TRACKS              Primary> 1           Secondary> 1
     CI Size> 4096

  Index name> CBL.SELCOPY.DEMO1.KSDS.INDEX
     Volumes> CBLM04                                     Ordered> N
  Space type> TRACKS              Primary> 1           Secondary> 1
     CI Size> 512                   Imbed> N           Replicate> N
```

Figure 121. Define VSAM KSDS window.

Menu Bar Items

Define
Start the VSAM object definition. (Foreground)

Job
Creates and edits the IDCAMS DEFINE statement including job control ready for submission to batch (See CBLe command SUBMIT.)

AMS

Opens a CBLe edit view containing generated AMS command syntax to perform the IDCAMS DEFINE. Execute by placing the cursor on the first line of the command and hitting <PF4> The command may be copied to the user's HOME command centre for future reference.

Help

Open the help window for VSAM elements definition.

DB2 Utilities

SELCOPY/i DB2 facilities are incorporated within the SELCOPY Product Suite base product and do not require any additional licensing over and above the SELCOPY product key.

SELCOPY/i DB2 provides a suite of tools to assist working with DB2 data and objects. Command syntax and panels provide functions that include:

- Edit and Browse of DB2 table data in a SELCOPY/i Structured Data Edit (SDE) view.
- List, Create, Drop and Alter of DB2 objects. (Tables, Indexes, etc.)
- Interactive Execution of DB2 commands and SQL statements.
- Generation of JCL for SQL statement execution and stand-alone DB2 utilities.

SELCOPY/i users may connect to any local DB2 subsystem for which the SELCOPY/i DB2 plan has been bound and the user granted EXECUTE authority. For successful operation, users must also be granted SELECT access to the subsystem's DB2 catalog tables and, if configured, READ access to the SELCOPYI.DB2 SAF resource. See the "*SELCOPY Product Suite Customisation Guide*" for details on enabling SELCOPY/i DB2.

Multiple connections to one or more local DB2 subsystems may exist in the same SELCOPY/i session. Each connection has its own audit setting (on or off) and audit log data set. The DB2 susbsystem name is displayed in the title bar of any SELCOPY/i DB2 window (panel or SDE edit view) to which that window relates.

All DB2 related features may be accessed via the suite of SELCOPY/i DB2 interactive panel windows (window class WINWIPO0) that are invoked by selecting the "DB2" drop-down menu item of the "File" main menu, or by executing the DB2 CLI command. Note that SELCOPY/i panels are window objects within SELCOPY/i and should not be confused with ISPF panels.

DB2 Primary Option Menu

The DB2 Primary Option Menu panel (ZZS2PRIM) is an interactive panel window providing the entry point to all SELCOPY/i DB2 panels, encompassing the DB2 functionality available in SELCOPY/i.

This DB2 panel is the first in a hierarchical chain of DB2 panels (menus, functions, lists) that are opened thereafter. Select an item from the menu of DB2 related tasks to open the relevant DB2 task panels. Note that, although part of the same DB2 panel hierarchy, these panels are not owned by DB2 primary options menu panel. Window Focus may be returned to the primary options panel to select another branch of DB2 task panels in the same hierarchy without having to exit existing DB2 panels.

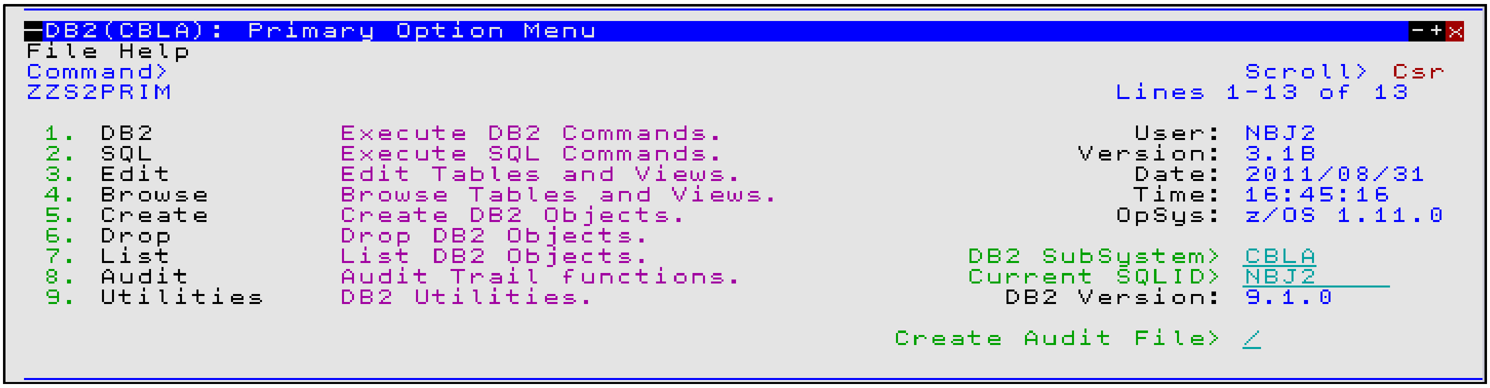
On selecting an item from this menu panel, or if a PFKey/<Enter> is actioned, an attempt is made to connect to the DB2 subsystem specified in the DB2 SubSystem field. If the value in this field is subsequently changed, then a new DB2 hierarchy of DB2 panels is started and a connection is made to the new DB2 subsystem. If no other panel exists in the DB2 hierarchy belonging to the previously specified DB2 subsystem, then connection to that subsystem is dropped before the new connection is made.

The name of the connected DB2 subsystem is displayed in parentheses in the window title bar of the DB2 primary options menu and therafter in the title bars of DB2 panels and SDE edit views opened in the DB2 panel hierarchy.

Although DB2 table SDE edit views may be opened via the DB2 panels, these are not included as part of the DB2 panel hierarchy. This is because a separate DB2 connection is performed and a separate audit file is maintained for each edited results table allowing updates made to a DB2 table to be isolated from other DB2 tasks (including updates made to other DB2 tables.)

To work with multiple DB2 subsystems concurrently, the command DB2 SSN=*name* may be executed to open multiple DB2 primary option menu panels each connected to different DB2 subsystems. Each invocation of a new DB2 primary option menu panel constitutes another hierarchy of DB2 related panels.

Note that if a DB2 primary option menu panel is already open for the SSN=*name*, then no new connection is made and the DB2 panel hierarchy that exists for that SSN is used instead.



```
 DB2(CBLA): Primary Option Menu                                              - + x
File Help
Command>                                                               Scroll> Csr
ZZS2PRIM                                                            Lines 1-13 of 13

 1. DB2          Execute DB2 Commands.                               User: NBJ2
 2. SQL          Execute SQL Commands.                            Version: 3.1B
 3. Edit         Edit Tables and Views.                              Date: 2011/08/31
 4. Browse       Browse Tables and Views.                            Time: 16:45:16
 5. Create       Create DB2 Objects.                               OpSys: z/OS 1.11.0
 6. Drop         Drop DB2 Objects.
 7. List         List DB2 Objects.                         DB2 SubSystem> CBLA
 8. Audit        Audit Trail functions.                    Current SQLID> NBJ2
 9. Utilities    DB2 Utilities.                              DB2 Version: 9.1.0

                                                    Create Audit File> /
```

Figure 122. DB2 Primary Option Menu Panel.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`

Open the general help for the DB2 Primary Option menu panel.

Options

1. DB2	Execute DB2 Commands
2. SQL	Execute SQL Statements
3. Edit	Edit Tables and Views
4. Browse	Browse Tables and Views
5. Create	Create DB2 Objects
6. Drop	Drop DB2 Objects
7. List	List DB2 Objects
8. Audit	Audit Trail Functions

Panel Input/Output Fields

`User:`

An output field displaying the user's userid.

`Version:`

An output field displaying the version of SELCOPY/i.

`Date:`

An output field displaying the current date.

`Time:`

An output field displaying the current time.

`OpSys:`

An output field displaying the operating system release.

`DB2 SubSystem>`

An input field identifying the DB2 subsystem to which a connection will be made. SELCOPY/i DB2 functions and panels will operate on objects defined in this subsystem. A connection will not be attempted until <Enter> is hit or the panel window is repainted (e.g. as a result of actioning a PFKey.)

The ZZS2PRIM internal field name for DB2 Subsystem is **SSN**.

`Current SQLID>`

An input field that sets the SQL authorisation ID for this particular hierarchy of DB2 panels' access to DB2. This value is the initial value of the DB2 special register CURRENT SQLID that is provided on the DB2 connection. See the IBM publication *"DB2 SQL Reference"* for further information on CURRENT SQLID and its usage with dynamically prepared SQL statements.

Changing the contents of this input field following connection will change the CURRENT SQLID for subsequent functions executed via panels in this DB2 panel hierarchy. To do this, the user requires appropriate DB2 authorisation to use the new value. (See the IBM publication *"DB2 Administration Guide"* for further information.)

The default value for Current SQLID is the user's TSO or SELCOPY/i VTAM logon id.

The ZZS2PRIM internal field name for Current SQLID is **SQLID**.

`DB2 Version:`

An output field displaying the version of DB2 for the connected DB2 subsystem. Note that DB2 version 9 is the earliest release supported by SELCOPY/i DB2.

`Create Audit File>`

An option check box that indicates that SELCOPY/i DB2 auditing will occur for actions performed in this DB2 panel hierarchy. Note that auditing of DB2 table edit views is managed separately and is not affected by this check box setting.

If Create Audit File is selected, an audit log file will be allocated immediately before attempting to connect to the DB2 subsystem and closed when the connection is dropped.

See Audit Trail Functions for details of SELCOPY/i DB2 auditing.

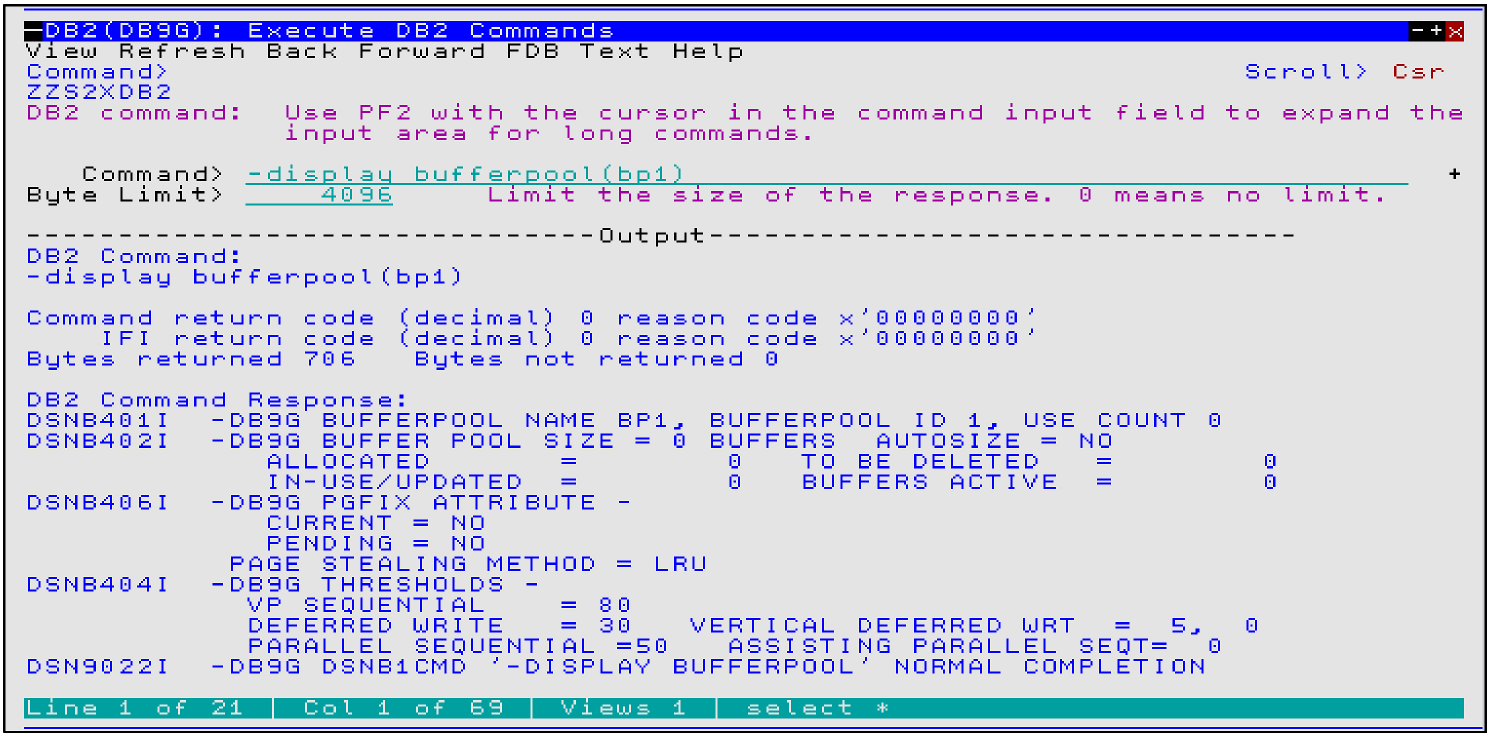
Execute DB2 Commands

The Execute DB2 Commands panel (ZZS2XDB2) is an interactive panel window, opened on selection of option 1. in the DB2 Primary options menu or on execution of the DCMD command.

This panel provides facility to execute DB2 and related commands to the connected DB2 subsystem and view the command output. (See the IBM publication *"DB2 Command Reference"* for further information.) The individual user must have the required level of authority in order to successfully execute a DB2 command.

Output from the DB2 command execution is displayed in a scrollable list window within the panel. The list consists of a single column with header "Output".

The Output data reports the DB2 command executed; the return and reason code received on execution of the DB2 command; the instrumentation facility interface (IFI) return and reason code; and the number of bytes returned/not returned. If number of bytes **not** returned is greater than zero, then this value indicates the amount of additional buffer space required to display the complete output from the command.



```
DB2(DB9G): Execute DB2 Commands
View Refresh Back Forward FDB Text Help
Command>                                                      Scroll> Csr
ZZS2XDB2
DB2 command:  Use PF2 with the cursor in the command input field to expand the
              input area for long commands.

   Command> -display bufferpool(bp1)                                          +
Byte Limit>     4096      Limit the size of the response. 0 means no limit.

------------------------------Output------------------------------------
DB2 Command:
-display bufferpool(bp1)

Command return code (decimal) 0 reason code x'00000000'
    IFI return code (decimal) 0 reason code x'00000000'
Bytes returned 706    Bytes not returned 0

DB2 Command Response:
DSNB401I  -DB9G BUFFERPOOL NAME BP1, BUFFERPOOL ID 1, USE COUNT 0
DSNB402I  -DB9G BUFFER POOL SIZE = 0 BUFFERS  AUTOSIZE = NO
             ALLOCATED       =        0    TO BE DELETED    =         0
             IN-USE/UPDATED  =        0    BUFFERS ACTIVE   =         0
DSNB406I  -DB9G PGFIX ATTRIBUTE -
             CURRENT = NO
             PENDING = NO
           PAGE STEALING METHOD = LRU
DSNB404I  -DB9G THRESHOLDS -
            VP SEQUENTIAL    = 80
            DEFERRED WRITE   = 30    VERTICAL DEFERRED WRT  =  5,  0
            PARALLEL SEQUENTIAL =50    ASSISTING PARALLEL SEQT=  0
DSN9022I  -DB9G DSNB1CMD '-DISPLAY BUFFERPOOL' NORMAL COMPLETION

Line 1 of 21 | Col 1 of 69 | Views 1 | select *
```

Figure 123. Execute DB2 Commands Panel.

Menu Bar Items

See List Window menu for description of menu bar items.

Field Entries

Command>

An input field in which the DB2 command is entered.

The ZZS2XDB2 internal field name for Command is **DB2CMD**.

Byte Limit>

An input field defining the maximum size of the DB2 command output data buffer.

If set to 0 (zero), then there is no limit to the output buffer size.

Where the length of data returned by the command exceeds the output buffer size, then error message ZZSX016W is returned indicating the number of bytes of output data returned, and number of bytes not returned by the command.

The ZZS2XDB2 internal field name for Byte Limit is **LIMIT**.

Execute SQL Statements

The Execute SQL Statements panel (ZZS2XSQL) is an interactive panel window, opened on selection of option 2. in the DB2 Primary options menu or on execution of the DSQL command.

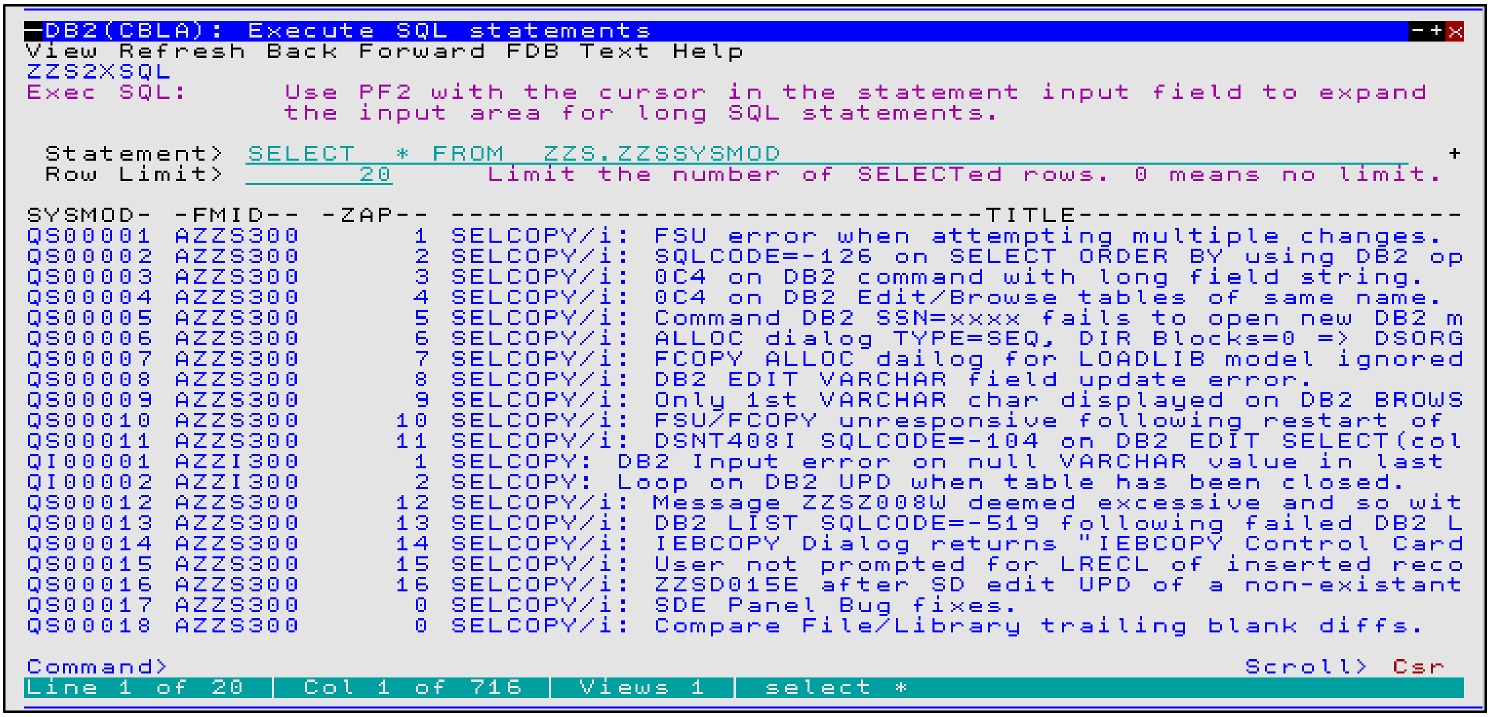
This panel provides facility to execute SQL statements to the connected DB2 subsystem and view the statement output. (See the IBM publication *"SQL Reference"* for further information.) The individual user must have the required level of authority in order to successfully execute a SQL statement.

Note that, whereas the Execute SQL Statements panel uses the SELCOPY/i DB2 plan (default CBLPLAN1) to execute the prepared SQL statement, the DB2 Dynamic SQL window may be used to perform the same operation using the SELCOPY batch program DB2 plan (default CBLPLAN0). This alternative method of executing DB2 SQL statements has the additional benefit of being able to process one or more SQL statements provided via an input contol file.

Output from the SQL statement execution is displayed in a scrollable list window within the panel.

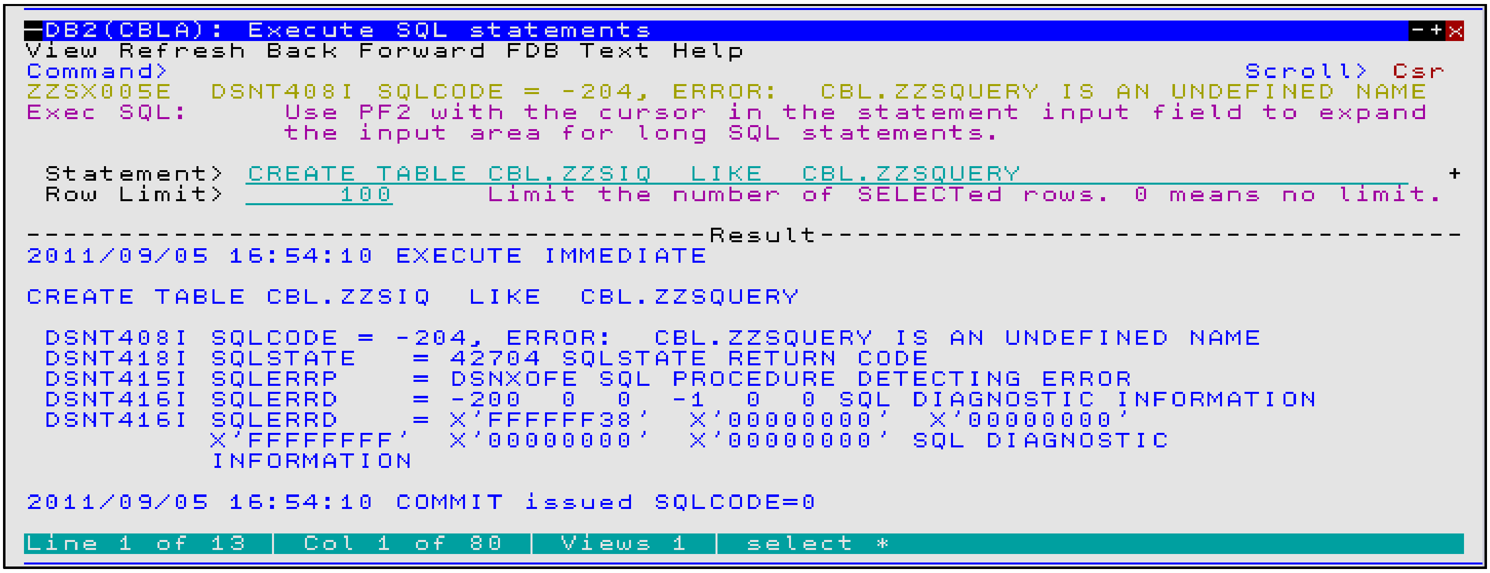
The format of the list output depends on the type of SQL statement executed. Successful execution of SQL query (SELECT) statements will display the selected results table columns and rows. The list column headers are the results table column names. All other SQL statements display a report of SQL messages detailing successful or unsuccessful execution. The message text output are rows of a list with the single column header, "Result".

See the IBM publications *"DB2 Messages"* and *"DB2 Codes"* for DSN prefixed messages and SQL error codes.



```
 DB2(CBLA): Execute SQL statements                                        -+x
View Refresh Back Forward FDB Text Help
ZZS2XSQL
Exec SQL:      Use PF2 with the cursor in the statement input field to expand
               the input area for long SQL statements.

 Statement> SELECT  * FROM  ZZS.ZZSSYSMOD                                    +
 Row Limit>        20     Limit the number of SELECTed rows. 0 means no limit.

SYSMOD- -FMID-- -ZAP-- -----------------------------TITLE---------------------
QS00001 AZZS300      1 SELCOPY/i: FSU error when attempting multiple changes.
QS00002 AZZS300      2 SELCOPY/i: SQLCODE=-126 on SELECT ORDER BY using DB2 op
QS00003 AZZS300      3 SELCOPY/i: 0C4 on DB2 command with long field string.
QS00004 AZZS300      4 SELCOPY/i: 0C4 on DB2 Edit/Browse tables of same name.
QS00005 AZZS300      5 SELCOPY/i: Command DB2 SSN=xxxx fails to open new DB2 m
QS00006 AZZS300      6 SELCOPY/i: ALLOC dialog TYPE=SEQ, DIR Blocks=0 => DSORG
QS00007 AZZS300      7 SELCOPY/i: FCOPY ALLOC dailog for LOADLIB model ignored
QS00008 AZZS300      8 SELCOPY/i: DB2 EDIT VARCHAR field update error.
QS00009 AZZS300      9 SELCOPY/i: Only 1st VARCHAR char displayed on DB2 BROWS
QS00010 AZZS300     10 SELCOPY/i: FSU/FCOPY unresponsive following restart of
QS00011 AZZS300     11 SELCOPY/i: DSNT408I SQLCODE=-104 on DB2 EDIT SELECT(col
QI00001 AZZI300      1 SELCOPY: DB2 Input error on null VARCHAR value in last
QI00002 AZZI300      2 SELCOPY: Loop on DB2 UPD when table has been closed.
QS00012 AZZS300     12 SELCOPY/i: Message ZZSZ008W deemed excessive and so wit
QS00013 AZZS300     13 SELCOPY/i: DB2 LIST SQLCODE=-519 following failed DB2 L
QS00014 AZZS300     14 SELCOPY/i: IEBCOPY Dialog returns "IEBCOPY Control Card
QS00015 AZZS300     15 SELCOPY/i: User not prompted for LRECL of inserted reco
QS00016 AZZS300     16 SELCOPY/i: ZZSD015E after SD edit UPD of a non-existant
QS00017 AZZS300      0 SELCOPY/i: SDE Panel Bug fixes.
QS00018 AZZS300      0 SELCOPY/i: Compare File/Library trailing blank diffs.

Command>                                                           Scroll> Csr
Line 1 of 20 | Col 1 of 716 | Views 1 | select *
```

Figure 124. Execute SQL Statements Panel - SQL Query.



```
 DB2(CBLA): Execute SQL statements                                        -+x
View Refresh Back Forward FDB Text Help
Command>                                                           Scroll> Csr
ZZSX005E  DSNT408I SQLCODE = -204, ERROR:  CBL.ZZSQUERY IS AN UNDEFINED NAME
Exec SQL:      Use PF2 with the cursor in the statement input field to expand
               the input area for long SQL statements.

 Statement> CREATE TABLE CBL.ZZSIQ  LIKE  CBL.ZZSQUERY                       +
 Row Limit>       100     Limit the number of SELECTed rows. 0 means no limit.

-------------------------------------Result-----------------------------------
2011/09/05 16:54:10 EXECUTE IMMEDIATE

CREATE TABLE CBL.ZZSIQ  LIKE  CBL.ZZSQUERY

 DSNT408I SQLCODE = -204, ERROR:  CBL.ZZSQUERY IS AN UNDEFINED NAME
 DSNT418I SQLSTATE   = 42704 SQLSTATE RETURN CODE
 DSNT415I SQLERRP    = DSNXOFE SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD    = -200  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD    = X'FFFFFF38'  X'00000000'  X'00000000'
          X'FFFFFFFF'  X'00000000'  X'00000000' SQL DIAGNOSTIC
          INFORMATION

2011/09/05 16:54:10 COMMIT issued SQLCODE=0

Line 1 of 13 | Col 1 of 80 | Views 1 | select *
```

Figure 125. Execute SQL Statements Panel.

Menu Bar Items

See List Window menu for description of menu bar items.

Field Entries

Statement>

An input field in which the SQL statement is entered.

The ZZS2XSQL internal field name for Statement **SQLCMD**.

Row Limit>

Limit the number of rows to selected by an SQL query (SELECT) statement. Once the limit threshold has been reached, no further attempt is made to retrieve selected rows of data.

If set to 0 (zero), then there is no limit to the number of rows retrieved from the results table.

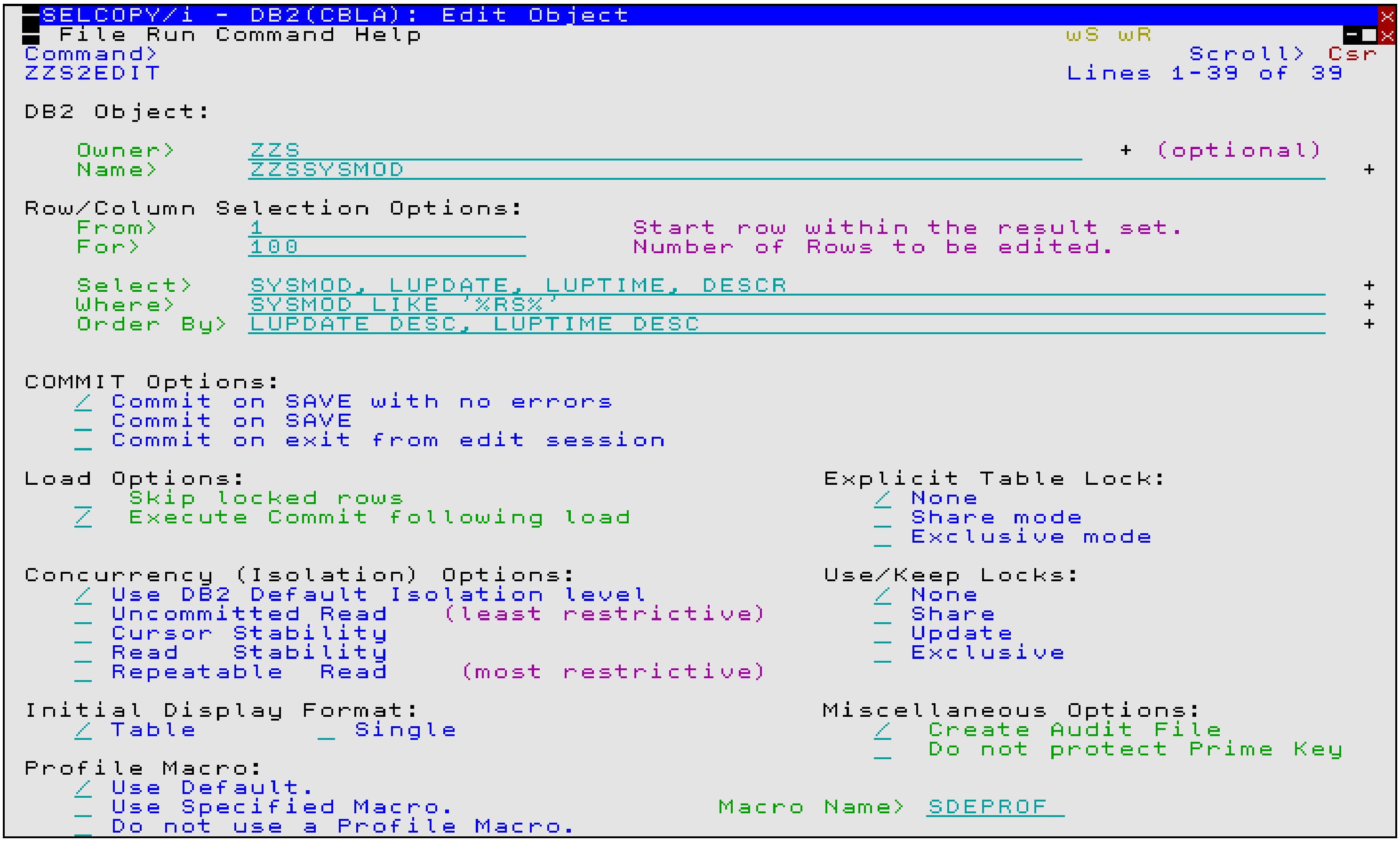
The ZZS2XSQL internal field name for Row Limit is **LIMIT**.

Edit Tables and Views

The Edit Object panel (ZZS2EDIT) is an interactive panel window, opened on selection of option 3. in the DB2 Primary options menu or on issuing prefix command "E" in a DB2 table/view/alias/synonym list.

This panel allows the user to configure DB2 table edit and data management options prior to loading rows from the DB2 results table and presenting the column data in an SDE edit view. General features of the SDE editor and features specific to SDE DB2 table edit views are documented in detail in the SELCOPY/i Structured Data Editor (SDE) manual.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the table edit. Prior to editing the table data, SELCOPY/i will open a new connection to the DB2 subsystem specified by the DB2 Primary option menu panel and, if selected, will allocate a new audit log data set. This isolates actions specific to the DB2 table edit from all other DB2 actions and table editing performed on the subsystem.



```
 SELCOPY/i - DB2(CBLA): Edit Object
  File Run Command Help                                          wS wR
Command>                                                                Scroll> Csr
ZZS2EDIT                                                           Lines 1-39 of 39

DB2 Object:

   Owner>     ZZS                                              + (optional)
   Name>      ZZSSYSMOD                                                          +

Row/Column Selection Options:
   From>      1                       Start row within the result set.
   For>       100                     Number of Rows to be edited.

   Select>    SYSMOD, LUPDATE, LUPTIME, DESCR                                    +
   Where>     SYSMOD LIKE '%RS%'                                                 +
   Order By>  LUPDATE DESC, LUPTIME DESC                                         +

COMMIT Options:
   / Commit on SAVE with no errors
   _ Commit on SAVE
   _ Commit on exit from edit session

Load Options:                                 Explicit Table Lock:
      Skip locked rows                           / None
   /  Execute Commit following load               _ Share mode
                                                 _ Exclusive mode

Concurrency (Isolation) Options:              Use/Keep Locks:
   / Use DB2 Default Isolation level             / None
   _ Uncommitted Read    (least restrictive)     _ Share
   _ Cursor Stability                            _ Update
   _ Read   Stability                            _ Exclusive
   _ Repeatable  Read    (most restrictive)

Initial Display Format:                       Miscellaneous Options:
   / Table         _ Single                      /  Create Audit File
                                                 _  Do not protect Prime Key
Profile Macro:
   / Use Default.
   _ Use Specified Macro.                 Macro Name> SDEPROF
   _ Do not use a Profile Macro.
```

Figure 126. DB2 Edit Object Panel.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the table edit for the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing the SDE EDIT command syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

Help

Open the general help for the DB2 Edit Object panel.

Field Entries

DB2 Object:

Fields that together identify the DB2 table or view in the current location which is to be edited.

Owner>

The owner of the required table or view.

Name>

The name of the required table or view.

Row/Column Selection Options:

Fields that together identify the DB2 results table rows and columns to be edited and the order in which they appear.

From>

Specifies the row number of the first row to be displayed for edit in the DB2 results table returned by the SQL query. This row becomes row 1 within the SDE window edit view. Rows that occur before this row number are not included within the edit session.
Default is row 1.

For>

Specifies the maximum number of rows to be displayed from the DB2 results table returned by the SQL query. Rows that fall outside the range of rows identified by the From and For fields are not included within the edit session.
Default is to display all selected rows.

Select>

Specifies a DB2 SQL SELECT clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL select clause.

Where>

Specifies a DB2 SQL WHERE clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL where clause.

Order By>

Specifies a DB2 SQL ORDER BY clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL order-by clause.

COMMIT Options:

Identifies when a DB2 COMMIT should be performed for changed table data. Mutually exclusive options are as follow:

◊ **Commit on SAVE with no errors**
COMMIT when SAVE is executed without errors. This option is default.

◊ **Commit on SAVE**
COMMIT when SAVE is executed regardless of any save errors.

◊ **Commit on exit from edit session**
COMMIT only when closing the last edit view of the table data.

`Load Options:`

Options relating to actions performed at load of results table rows.

`Skip locked rows`

This option field is ignored unless an isolation level of Cursor Stability (CS) or Read Stability (RS) is in effect. Skip locked rows indicates that any selected rows that are already locked by another process should be skipped and are not included in the edit display. See "*DB2 SQL Reference*" for details on the SKIP LOCKED DATA clause.
Default is to allow display of locked rows whenever possible.

`Execute Commit following load`

Indicates that a COMMIT is to be performed following the initial load of the results table rows selected for edit, thus releasing all DB2 table locks performed during load of the data. This includes any explicit table locks that may have been applied.
Default is **not** to perform a COMMIT following load of the table rows.

`Explicit Table Lock:`

Applicable to edit of DB2 base tables only, indicates whether or not a DB2 SQL LOCK TABLE command should be executed before loading rows from the table and, if so, the mode of lock to be performed.

Note that **caution** should be taken when selecting a table locking option as other users and applications may be prevented from accessing the table. If the table is locked then the lock is held until the next COMMIT is actioned. See "*DB2 SQL Reference*" for details on the effects of LOCK TABLE options.

Mutually exclusive options are as follow:

◊ **None**
No explict table locking prior to load. This option is default.

◊ **Share mode**
Share mode table locking prevents anything other than read-only operations from being performed on the table whilst the lock is in effect.

◊ **Exclusive mode**
Unless a process is running with an isolation level of Uncommitted Read (UR) in which case read-only (dirty read) operations may be performed, exclusive mode table locking prevents another process from performing any operation on the table whilst the lock is in effect.

`Concurrency (Isolation) Options:`

Specifies a DB2 SQL isolation-clause to be included in the prepared SQL select statement. See "*DB2 SQL Reference*" for details on the SQL Isolation clause and "*DB2 Performance Monitoring and Tuning Guide*" for details on the effects of isolation level on concurrency and protection of DB2 table data.

Mutually exclusive options are as follow:

◊ **Use DB2 Default Isolation level**
The default isolation level as set by the BIND of the SELCOPY/i DB2 package and plan. (See sample job ZZSDB2B which sets default isolation of Cursor Stability.)

◊ **Uncommitted Read**
Uncommitted Read (UR) acquires few locks but may result in edit of uncommitted data.

◊ **Cursor Stability**
Cursor Stability (CS) allows maximum concurrency with data integrity.

This isolation method will realease a lock for a row after it has been read. If an attempt is made to update or delete a row (on save), then the row is locked, the predicate re-evaluated to ensure that it still qualifies for the results table and only then, if values match, is the row updated or deleted.

◊ **Read Stability**
Read Stability (RS) locks only those rows that satisfy the results table predicate, releasing the locks only when a COMMIT is performed. Rows that do not satisfy the predicate are eligible for update or delete by other applications. New rows may also be inserted by other applications.

The mode of lock performed on the rows is indicated by the Use/Keep Locks options.

◊ **Repeatable Read**
Repeatable Read (RR) is most restrictive since it prevents other applications from performing update, insert or delete even on rows that do not satisfy the results table predicate. Locks on all accessed rows are held until a COMMIT is performed.

The mode of lock performed on the rows is indicated by the Use/Keep Locks options.

Use/Keep Locks:

Applicable only if isolation level Read Stability or Repeatable Read is selected, this option identifies the mode of lock to be used and kept on locked pages and rows.

Mutually exclusive options are as follow:

◊ **None**
Use and keep a row using the default lock mode.

◊ **Share**
Concurrent processes can read but not change the locked row and may acquire SHARE or UPDATE mode locks on the row.

◊ **Update**
Concurrent processes can acquire a SHARE mode lock on the row and may only read the data without acquiring a page or row lock.

◊ **Exclusive**
Concurrent processes cannot acquire any mode of lock on the row.

Initial Display Format:

Identifies the initial display format of the DB2 table rows in the SDE window view.

Mutually exclusive options are as follow:

◊ **Table**
Display the DB2 table rows in tabular multi record view.

◊ **Single**
Display the DB2 table rows one at a time in single record view.

Miscellaneous Options:

Miscellanneous eidt options.

Create Audit File

Open a new SELCOPY/i DB2 audit data set to record changes to the edited table made during this edit session. See DB2 Audit Trail Functions for details.

Do not protect Prime Key

If selected, specifies that data occupying columns that comprise the table's primary key is eligible for edit and update.

Profile Macro:

Indicates whether or not an SDE data edit profile macro is to be executed when the SDE window view is opened and, if so, the macro to be executed.

Mutually exclusive options are as follow:

◊ **Use Default.**
Use the default SDE edit profile macro (SDEPROF).

◊ **Use Specified Macro.**
Use an SDE edit profile macro with macro name specified by the Macro Name> field.

◊ **Do not use a Profile Macro.**
No SDE edit profile macro is to be executed - all edit options are default.

Macro Name>

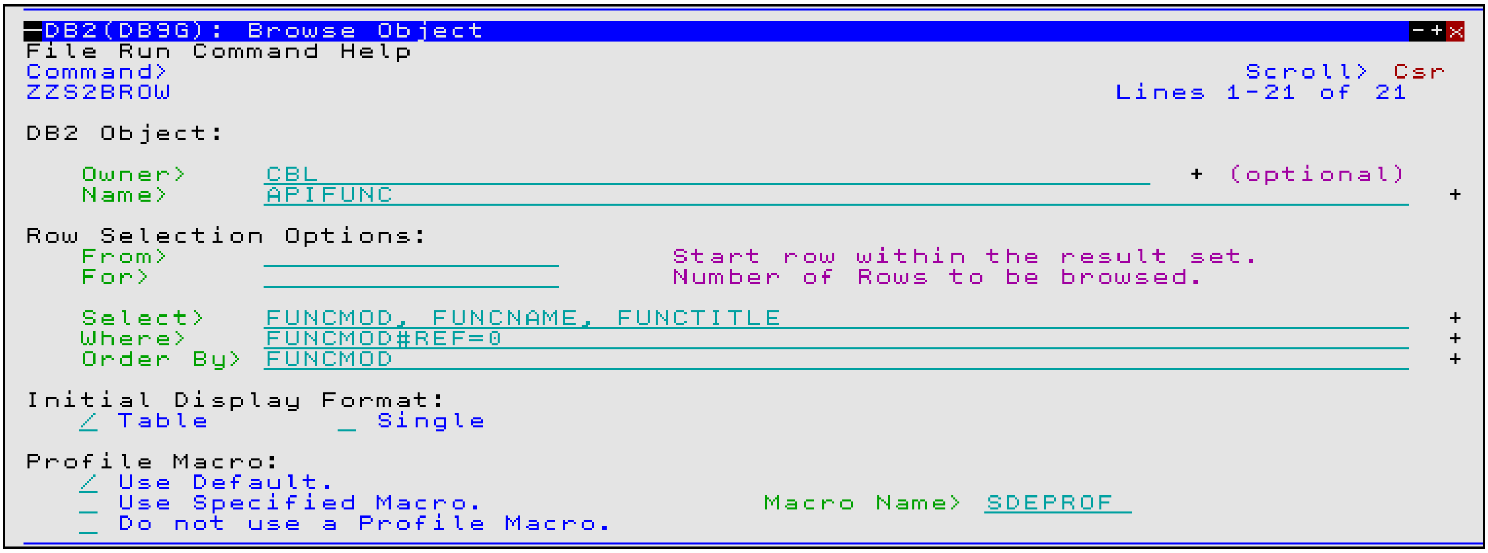
Applicable only if option Use Specified Macro is selected. this field names the SDE edit profile macro to be executed when the table is edited.

Browse Tables and Views

The Browse Object panel (ZZS2BROW) is an interactive panel window, opened on selection of option 4. in the DB2 Primary options menu or on issuing prefix command "B" in a DB2 table/view/alias/synonym list.

This panel allows the user to configure DB2 table browse options prior to loading rows from the DB2 results table and presenting the column data in an SDE window view. General features of the SDE editor and features specific to SDE DB2 table browse views are documented in detail in the SELCOPY/i Structured Data Editor (SDE) manual.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the table browse.



```
DB2(DB9G): Browse Object
File Run Command Help
Command>                                                                    Scroll> Csr
ZZS2BROW                                                               Lines 1-21 of 21

DB2 Object:

   Owner>     CBL                                                  + (optional)
   Name>      APIFUNC                                                             +

Row Selection Options:
   From>      ________________        Start row within the result set.
   For>       ________________        Number of Rows to be browsed.

   Select>    FUNCMOD, FUNCNAME, FUNCTITLE                                        +
   Where>     FUNCMOD#REF=0                                                       +
   Order By>  FUNCMOD                                                             +

Initial Display Format:
   / Table           _ Single

Profile Macro:
   / Use Default.
   _ Use Specified Macro.                       Macro Name> SDEPROF
   _ Do not use a Profile Macro.
```

Figure 127. DB2 Browse Object Panel.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the table browse for the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view of a temporary data set containing the SDE BROWSE command syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

Help

Open the general help for the DB2 Browse Object panel.

Field Entries

DB2 Object:

Fields that together identify the DB2 table or view in the current location which is to be browsed.

Owner>

The owner of the required table or view.

Name>

The name of the required table or view.

Row Selection Options:

Fields that together identify the DB2 results table rows and columns to be browsed and the order in which they appear.

From>

Specifies the row number of the first row to be displayed for browse in the DB2 results table returned by the SQL query. This row becomes row 1 within the SDE window view. Rows that occur before this row number are not included within the browse session.
Default is row 1.

For>

Specifies the maximum number of rows to be displayed from the DB2 results table returned by the SQL query. Rows that fall outside the range of rows identified by the From and For fields are not included within the browse session.
Default is to display all selected rows.

Select>

Specifies a DB2 SQL SELECT clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL select clause.

Where>

Specifies a DB2 SQL WHERE clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL where clause.

Order By>

Specifies a DB2 SQL ORDER BY clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL order-by clause.

Initial Display Format:

Identifies the initial display format of the DB2 table rows in the SDE window view.

Mutually exclusive options are as follow:

◊ **Table**
Display the DB2 table rows in tabular multi record view.

◊ **Single**
Display the DB2 table rows one at a time in single record view.

Profile Macro:

Indicates whether or not an SDE data edit profile macro is to be executed when the SDE window view is opened and, if so, the macro to be executed.

Mutually exclusive options are as follow:

◊ **Use Default.**
Use the default SDE edit profile macro (SDEPROF).

◊ **Use Specified Macro.**
Use an SDE edit profile macro with macro name specified by the Macro Name> field.

◊ **Do not use a Profile Macro.**
No SDE edit profile macro is to be executed - all edit options are default.

Macro Name>

Applicable only if option Use Specified Macro is selected. this field names the SDE edit profile macro to be executed when the table is edited.

Create DB2 Objects

The Create DB2 Object panel (ZZS2C000) is an interactive panel window, opened on selection of option 5. in the DB2 Primary options menu.

This panel allows the user to select the type of object to be created by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Note that successful creation of some DB2 objects is dependent upon the version of DB2 used by the connected DB2 system to which this panel applies. An SQL error message will occur if an SQL CREATE parameter field is used which is unsupported by the DB2 version. Similarly, successful creation of individual DB2 objects is also dependent upon the user's level of authority or granted privileges. Please refer to the relevant edition of the *"z/OS SQL Reference"*.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Help

Open the general help for the Create DB2 Objects option menu panel.

Options

1. Storage group
2. Work File Database
3. User Database
4. View
5. Alias
6. Synonym
7. Type
8. Trigger
9. Sequence
10. Role

Create Storage Group

The Create Storage Group panel (ZZS2CSG0) is an interactive panel window, opened on selection of option 1. in the Create DB2 Object options menu. and may be used to create a new storage group in the current server.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE STOGROUP statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE STOGROUP generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE STOGROUP syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE STOGROUP statement syntax generated for the selected input field values.

Help

Open the general help for the Create Storage Group panel.

Field Entries

Storage Group Name>

The name of a new storage group to be created at the current server. Maximum length of a storage group name is 128 characters.

Volume Serial Numbers>

A volume ID or a comma separated list of volume IDs to be assigned to the storage group.

If a data set associated with the storage group is to be SMS managed, '*' (asterisk) may be specified as one of the volume IDs in order to allow SMS to select volumes as appropriate. Use of SMS is highly recommended rather than using DB2 to allocate data to specific volumes which would require non-SMS usage or an SMS Storage Class with guaranteed space. Assigning an SMS Storage Class with guaranteed space is not recommended as it reduces the benefits of SMS allocation.

A volume ID may be specified once only and the maximum number of volumes IDs in a storage group is 133. If one or more of DATACLAS, MGMTCLAS or STORCLAS is specified, then an entry in this input field is optional and so volume selection controlled by SMS.
This parameter field corresponds to SQL CREATE STOGROUP parameter VOLUMES

Catalog name or ALIAS>

Specifies the ICF catalog in which DB2 data sets will be cataloged.

The ICF catalog name or catalog alias has a maximum length of 8 characters and its specification is mandatory.
This parameter field corresponds to SQL CREATE STOGROUP parameter VCAT.

Associated SMS Classes:
Applicable to SMS controlled volume selection only.

DATACLAS>
The name of the SMS data class to be associated with the DB2 storage group. (Maximum length 8 characters).
This parameter field corresponds to SQL CREATE STOGROUP parameter DATACLAS.

MGMTCLAS>
The name of the SMS management class to be associated with the DB2 storage group. (Maximum length 8 characters).
This parameter field corresponds to SQL CREATE STOGROUP parameter MGMTCLAS.

STORCLAS>
The name of the SMS storage class to be associated with the DB2 storage group. (Maximum length 8 characters).
This parameter field corresponds to SQL CREATE STOGROUP parameter STORCLAS.

Create Work File Database

The Create Work File Database panel (ZZS2CDBW) is an interactive panel window, opened on selection of option 2. in the Create DB2 Object options menu. Applicable only in a data sharing environment, this panel may be used to create the one and only work file database in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE DATABASE AS WORKFILE statement in the SELCOPY/i foreground.

Menu Bar Items

File
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run
Executes the SQL CREATE DATABASE generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command
Opens a text edit view for a temporary data set containing SQL command CREATE DATABASE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL
Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE DATABASE statement syntax generated for the selected input field values.

Help
Open the general help for the Create Work File Database panel.

Field Entries

Database Name>
The name of the new work file database.

The database name has a maximum length of 8 characters and must not be DSN%%%%% (where '%' is any single character).

Storage Group name>
The name of the default storage group to be used as required when allocating data sets for database table spaces and indexes.

The maximum length of a storage group name is 128 characters and, if not specified, defaults to SYSDEFLT.
This parameter field corresponds to SQL CREATE DATABASE parameter STOGROUP.

Create for Member>

Specifies the DB2 data sharing member name of the DB2 subsystem for which this work file database applies.

The maximum length of a member name is 8 characters and, if not specified, defaults to the member name of the current DB2 subsystem.
This parameter field corresponds to SQL CREATE DATABASE parameter FOR.

Buffer Pool names:

Default buffer pools to be used for table spaces and indexes created in the database.

Table Spaces>

Specifies the default buffer pool name to be used for table spaces.

Possible buffer pool names are BPn (n=0 to 49) which correspond to buffer pools of size 4K.

If not specified, the buffer pool specified for user data in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter BUFFERPOOL.

Indexes>

Specifies the default buffer pool name to be used for indexes.

Possible buffer pool names are BPn (n=0 to 49), BP8Kn, BP16Kn (n=0 to 9) and BP32K or BP32Kn (n=1 to 9).
These correspond to buffer pools of size 4K, 8K, 16K and 32K respectively.

If not specified, the buffer pool specified for user indexes in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter INDEXBP.

Create User Database

The Create User Database panel (ZZS2CDBU) is an interactive panel window, opened on selection of option 3. in the Create DB2 Object options menu. This panel may be used to create a user database in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE DATABASE statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE DATABASE generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE DATABASE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE DATABASE statement syntax generated for the selected input field values.

Help

Open the general help for the Create User Database panel.

Field Entries

Database Name>

The name of the new user database.

The database name has a maximum length of 8 characters and must not be DSN%%%%% (where '%' is any single character) and must not begin with DSNDB.

Storage Group name>

The name of the default storage group to be used as required when allocating data sets for database table spaces and indexes.

The maximum length of a storage group name is 128 characters and, if not specified, defaults to SYSDEFLT.
This parameter field corresponds to SQL CREATE DATABASE parameter STOGROUP.

Buffer Pool names:

Default buffer pools to be used for table spaces and indexes created in the database.

Table Spaces>

Specifies the default buffer pool name to be used for table spaces.

Possible buffer pool names are BPn (n=0 to 49), BP8Kn, BP16Kn (n=0 to 9) and BP32K or BP32Kn (n=1 to 9). These correspond to buffer pools of size 4K, 8K, 16K and 32K respectively.

If not specified, the buffer pool specified for **user data** in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter BUFFERPOOL.

Indexes>

Specifies the default buffer pool name to be used for indexes.

Possible buffer pool names are BPn (n=0 to 49), BP8Kn, BP16Kn (n=0 to 9) and BP32K or BP32Kn (n=1 to 9). These correspond to buffer pools of size 4K, 8K, 16K and 32K respectively.

If not specified, the buffer pool specified for **user indexes** in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter INDEXBP.

Database Encoding:

Default encoding scheme for table spaces created in the database. Mutually exclusive options are as follow:

◊ **Default**
Data must be encoded using the default CCSIDs as defined by the DEF ENCODING SCHEME value specified in the installation panel DSNTIPF.

◊ **EBCDIC**
Data must be encoded using the EBCDIC CCSIDs of the server.

◊ **ASCII**
Data must be encoded using the ASCII CCSIDs of the server.

◊ **UNICODE**
Data must be encoded using the UNICODE CCSIDs of the server.

Create View

The Create View panel (ZZS2CVI0) is an interactive panel window, opened on selection of option 4. in the Create DB2 Object options menu. This panel may be used to create a DB2 view on tables or views in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE VIEW statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE VIEW generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE VIEW syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE VIEW statement syntax generated for the selected input field values.

Help

Open the general help for the Create View panel.

Field Entries

View:

Fields that together identify the DB2 view name to be created at the current server location. The view name must not match the name of an existing table, view alias or synonym.

Owner>

The owner (schema) of the new view.
The view owner value has a maximum length of 128 characters.

Name>

The name of the new view.
The view name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE VIEW parameter *view-name*.

Full select

Specifies an SQL fullselect that defines the view. At any time, the view consists of the columns and rows that would result if the fullselect were executed.
This parameter field is mandatory and corresponds to SQL CREATE VIEW parameter *fullselect*.

Common table expression

Specifies a common table expression, for use with the full select part of the view definition.
This parameter field is optional and corresponds to SQL CREATE VIEW parameter WITH *common-table-expression*.

Column names

Specifies a comma separated list of column names that are used to name each column in the view. If specified, number of column names must match the number of columns returned by the results table of the fullselect.

If not specified, the columns names of the view inherit those of the results table generated by the fullselect.
This parameter field corresponds to SQL CREATE VIEW parameter *column-name*.

With Check Option:

Determines whether or not a check that an inserted or updated row conforms to the definition of the view is performed and, if so, the extent of the checking performed.

◊ **Default**
The definition of the view is **not** used to perform checks on unserted or updated rows.

◊ **Cascaded**
Updated and inserted rows must satisfy the search conditions of the view and all underlying views regardless of whether those underlying views were defined with the CHECK option.
This option corresponds to SQL CREATE VIEW parameter WITH CASCADED CHECK OPTION.

◊ **Local**
Updated and inserted rows must satisfy the search conditions of the view and all underlying views that have been defined with the CHECK option.
This option corresponds to SQL CREATE VIEW parameter WITH LOCAL CHECK OPTIONS.

Create Alias

The Create Alias panel (ZZS2CAL0) is an interactive panel window, opened on selection of option 5. in the Create DB2 Object options menu. This panel may be used to create in the current DB2 subsystem, a DB2 alias for a table or view.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE ALIAS statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE ALIAS generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE ALIAS syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE ALIAS statement syntax generated for the selected input field values.

Help

Open the general help for the Create Alias panel.

Field Entries

Alias:

Fields that together identify the DB2 alias name to be created at the current server location. The alias name must not match the name of an existing table, view alias or index.

Owner>

The owner (schema) of the new alias.
The alias owner value has a maximum length of 128 characters.

Name>

The name of the new alias.
The alias name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE ALIAS parameter *alias-name*.

Table or View:

Fields that together identify the DB2 table or view for which the alias is defined.

Location>

The DBMS location where the table or view is defined.
The table or view location name has a maximum length of 16 characters.

Owner>

The owner (schema) of the table or view.
The table or view owner name has a maximum length of 128 characters.

Name>

The SQL identifier name of the table or view.
The table or view name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE ALIAS parameter FOR *table-name* or FOR *view-name*.

Create Synonym

The Create Synonym panel (ZZS2CSY0) is an interactive panel window, opened on selection of option 6. in the Create DB2 Object options menu. This panel may be used to create a DB2 synonym for a table or view in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE SYNONYM statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE SYNONYM generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE SYNONYM syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE SYNONYM statement syntax generated for the selected input field values.

Help

Open the general help for the Create Synonym panel.

Field Entries

Synonym:

Specifies the DB2 synonym name to be created at the current server location. The synonym name must not match the name of an existing table, view, alias or synonym owned by the user.
The synonym name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE SYNONYM parameter *synonym*.

Table, View or Alias Name:

Fields that together identify the DB2 table, view or alias at the current server for which the synonym is defined.

Owner>

The owner (schema) of the table, view or alias.
The table, view or alias owner name has a maximum length of 128 characters.

Name>

The SQL identifier name of the table, view or alias.
The table, view or alias name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE SYNONYM parameter FOR *authorisation-name.table-name* or FOR *authorisation-name.view-name*.

Create Distinct Type

The Create Distinct Type panel (ZZS2CTY0) is an interactive panel window, opened on selection of option 7. in the Create DB2 Object options menu. This panel may be used to create a DB2 distinct type for a specific data type.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE TYPE statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE TYPE generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE TYPE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE TYPE statement syntax generated for the selected input field values.

Help

Open the general help for the Create Distinct Type panel.

Field Entries

Distinct Type:

Fields that together identify the DB2 distinct type name to be created at the current server location. The distinct type name must not match the name of an existing distinct type, built-in type, BOOLEAN or a system reserved keyword.

Schema>

The schema of the new distinct type.
The schema name has a maximum length of 128 characters.

Name>

The SQL identifier name of the distinct name.
The distinct name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE TYPE parameter *distinct-type-name*.

Select Both Source Field Type and Specific Data Type:

Fields that together identify the DB2 source built-in data type. Mutually exclusive options are as follow:

◊ **Signed Numeric Field Type**
Built-in type is numeric. Mutually exclusive options are as follow:

- **SMALLINT**
- **INTEGER**
- **BIGINT**
- **FLOAT SINGLE**
- **FLOAT DOUBLE**
- **DECFLOAT(16)**
- **DECFLOAT(34)**
- **DECIMAL**

Precision>

Applicable only to built-in data type DECIMAL, specifies the precision (total number of digits) supported by the decimal field. This value may be 1 to 31.

Scale>

Applicable only to built-in data type DECIMAL, specifies the scale (number of digits to the right of the decimal point) supported by the decimal field. This value may be 0 to value of the precision field.

- **String Field Type**
Built-in type is non-numeric data. Mutually exclusive options are as follow:

♦ **CHARACTER**
♦ **VARCHAR**
♦ **CLOB**

- **GRAPHIC**
- **VARGRAPHIC**
- **DBCLOB**
- **BINARY**
- **VARBINARY**
- **BLOB**

`Length>`

Specifies the fixed or maximum length of the data field. Maximum value for each data type is as follow:

CHARACTER	255
VARCHAR	Max row size - 8
CLOB	2147483647, 2097152K, 2048M or 2G
GRAPHIC	127
VARGRAPHIC	(Max row size - 2) / 2
DBCLOB	1073741823, 1048575K, 1024M or 1G
BINARY	255
VARBINARY	Max row size - 8
BLOB	2147483647, 2097152K, 2048M or 2G

`Data SubType:`

Applicable only to built-in data types CHARACTER, VARCHAR and CLOB, specifies the character data sub-type. Mutually exclusive options are as follow:

- **Default** - Use the default subtype for the CCSID encoding.
- **SBCS** - Single-byte data.
- **MIXED** - Mixed data.
- **BIT** - Bit data. (Not valid for CLOB).

`Data Encoding:`

Applicable only to built-in data types SBCS/MIXED CHARACTER or GRAPHIC, specifies the encoding scheme for the distinct type. Mutually exclusive options are as follow:

- **Default** - Default encoding scheme.
- **EBCDIC**
- **ASCII**
- **UNICODE**

- **Date/Time Field Type**
  Built-in type is a date, time or timestamp. Mutually exclusive options are as follow:

  - **DATE**
  - **TIME**
  - **TIMESTAMP**

- **Row Identifier Type**
  Built-in type is a row id. (**ROWID**)

Create Trigger

The Create Trigger panel (ZZS2CTR0) is an interactive panel window, opened on selection of option 8. in the Create DB2 Object options menu. This panel may be used to create a trigger in a schema and build a trigger package in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE TRIGGER statement in the SELCOPY/i foreground.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL CREATE TRIGGER generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

`Command`

Opens a text edit view for a temporary data set containing SQL command CREATE TRIGGER syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE TRIGGER statement syntax generated for the selected input field values.

Help

Open the general help for the Create Trigger panel.

Field Entries

Trgger:

Fields that together identify the DB2 trigger name to be created at the current server location. The trigger name must not match the name of an existing trigger.

Schema>

The schema of the new trigger.
The schema name has a maximum length of 128 characters.

Name>

The SQL identifier name of the trigger name.
The trigger name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE TRIGGER parameter *trigger-name*.

Order:

Identifies when the trigger will be implemented when invoked. Mutually exclusive options are as follow:

◊ **Before (No cascade)**
Action is triggered before any insert/delete/update action on the subject table. No other triggers are activated since the triggered action cannot include updates.
This option corresponds to SQL CREATE TRIGGER parameter NO CASCADE BEFORE.

◊ **After**
Action is triggered after any insert/delete/update action on the subject table.
This option corresponds to SQL CREATE TRIGGER parameter AFTER.

◊ **Instead**
Action is triggered instead of any insert/delete/update action on the subject view.
This option corresponds to SQL CREATE TRIGGER parameter INSTEAD OF.

Type:

Identifies the type of trigger. Mutually exclusive options are as follow:

◊ **Insert**
Action is triggered whenever there is an insert operation on the subject table.
This option corresponds to SQL CREATE TRIGGER parameter INSERT.

◊ **Update**
Action is triggered whenever there is an update operation on the subject view.
This option corresponds to SQL CREATE TRIGGER parameter UPDATE.

◊ **Delete**
Action is triggered whenever there is a delete operation on the subject table.
This option corresponds to SQL CREATE TRIGGER parameter DELETE.

Executed:

Specifies the conditions for which DB2 executes the triggered action. Mutually exclusive options are as follow:

◊ **For each row**
Action is triggered for each row of the subject table/view modified by the triggering operation.
This option corresponds to SQL CREATE TRIGGER parameter FOR EACH ROW.

◊ **For each statement**
Action is triggered once only by the triggering operation.
This option corresponds to SQL CREATE TRIGGER parameter FOR EACH STATEMENT.

Update Columns:

Applicable to trigger type UPDATE only, optionally specifies that activation of the trigger should be restricted only to operations that update one or more of these named columns. Column names must be comma separated.
This parameter field corresponds to SQL CREATE TRIGGER parameter UPDATE OF *column-name*.

Subject Table/View:
Fields that together identify the subject DB2 base table of a BEFORE or AFTER trigger definition, or the subject DB2 view of an INSTEAD OF trigger definition.

Owner>
The owner (schema) of the table or view.
The table or view owner name has a maximum length of 128 characters.

Name>
The SQL identifier name of the table or view.
The table or view name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE TRIGGER parameter ON *table-name* or ON *view-name*.

Triggered SQL:
Specifies a list of semi-colon separated SQL statements that are to be executed by the triggering action.
This parameter field is mandatory and corresponds to SQL CREATE TRIGGER parameters BEGIN ATOMIC *triggered-SQL-statement*; ... END

Search Conditions:
Specifies the condition that must be met for the triggered action to be executed.

If not specified, the triggered SQL ststaments will always be actioned.
This parameter field corresponds to SQL CREATE TRIGGER parameter WHEN (*search-condition*).

Referencing Old:
Optionally specifies the correlation names for the transition variables and the table names for the transition tables prior to any changes.

Correlation Name>
Identifies a name by which the triggered action can refer to values in the row before the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING OLD AS *correlation-name*.

Table identifier>
Identifies a name by which the triggered action can refer to the table before the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING OLD_TABLE AS *table-identifier*.

Referencing New:
Optionally specifies the correlation names for the transition variables and the table names for the transition tables following modifications.

Correlation Name>
Identifies a name by which the triggered action can refer to values in the row after the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING NEW AS *correlation-name*.

Table identifier>
Identifies a name by which the triggered action can refer to the table after the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING NEW_TABLE AS *table-identifier*.

Create Sequence

The Create Sequence panel (ZZS2CSQ0) is an interactive panel window, opened on selection of option 9. in the Create DB2 Object options menu. This panel may be used to create a DB2 sequence in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE SEQUENCE statement in the SELCOPY/i foreground.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL CREATE SEQUENCE generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

`Command`

Opens a text edit view for a temporary data set containing SQL command CREATE SEQUENCE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE SEQUENCE statement syntax generated for the selected input field values.

`Help`

Open the general help for the Create Sequence panel.

Field Entries

`Sequence:`

Fields that together identify the DB2 sequence name to be created at the current server location. The sequence name must not match the name of an existing sequence or one that is generated by DB2.

`Schema>`

The schema of the new sequence.
The schema name has a maximum length of 128 characters.

`Name>`

The SQL identifier name of the new sequence.
The sequence name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE SEQUENCE parameter *sequence-name*.

`Sequence Initialisation:`

Options relating to initial sequence values.

`Data Type>`

Specifies the data type used for the sequence value. Acceptable values are: SMALLINT, INTEGER, BIGINT or DECIMAL (scale zero).

If not specified, the default data type is INTEGER.
This parameter field corresponds to SQL CREATE SEQUENCE parameter AS *data-type*.

`Start Value>`

Specifies the first positive or negative value in the sequence.

If not specified, the start value is the MINVALUE for ascending sequences or the MAXVALUE for descending sequences.
This parameter field corresponds to SQL CREATE SEQUENCE parameter START WITH *numeric-constant*.

`Increment Value>`

Specifies the interval between successive values in the sequence. A positive value or 0 (zero) indicates an ascending sequence, a negative value indicates a descending sequence.

If not specified, the increment value is +1.
This parameter field corresponds to SQL CREATE SEQUENCE parameter INCREMENT BY *numeric-constant*.

Sequencing Options:

General options relating to the sequence.

Specify minimum

Activate this option as required by entering "/" in the selection field.

If not activated, no minimum value limit is set so that the default minimum value for a descending sequence is the minimum value for the data type.
This parameter field corresponds to SQL CREATE SEQUENCE parameter NO MINVALUE.

MIN Value>

Specifies the minimum value at which a descending sequence either cycles or stops generating values, or an ascending value cycles to when the maximum value is encountered.
This parameter field corresponds to SQL CREATE SEQUENCE parameter MINVALUE *numeric-constant*.

Specify maximum

Activate this option as required by entering "/" in the selection field.

If not activated, no explicit maximum value limit is set so that the default maximum value for an ascending sequence is the maximum value for the data type.
This parameter field corresponds to SQL CREATE SEQUENCE parameter NO MAXVALUE.

MAX Value>

Specifies the maximum value at which an ascending sequence either cycles or stops generating values, or a descending value cycles to when the minimum value is encountered.
This parameter field corresponds to SQL CREATE SEQUENCE parameter MAXVALUE *numeric-constant*.

Enable caching

Activate this option as required by entering "/" in the selection field.

If not activated, no pre-allocated values are kept in memory for faster access.
This parameter field corresponds to SQL CREATE SEQUENCE parameter NO CACHE.

Cache Value>

Specifies the maximum number of sequence values to be kept in cache memory.

The minimum value that may be specified is 2.
This parameter field corresponds to SQL CREATE SEQUENCE parameter CACHE *integer-constant*.

Cycle values

Activate this option as required by entering "/" in the selection field.

If activated, ascending sequence values will cycle to the minimum value when the maximum value threshold is encountered, and descending sequence values will cycle to the maximum value when the minimum value threshold is encountered. If not activated, no further sequence values are generated when minimum/maximum threshold limits are encountered.

This parameter field corresponds to SQL CREATE SEQUENCE parameter CYCLE and NO CYCLE.

Sequence in Order of request

Activate this option as required by entering "/" in the selection field.

If activated, sequence numbers will be generated in order of request. If not activated, sequence numbers do not need to be generated in order of request.

This parameter field corresponds to SQL CREATE SEQUENCE parameter ORDER and NO ORDER.

Create Role

The Create Role panel (ZZS2CRO0) is an interactive panel window, opened on selection of option 9. in the Create DB2 Object options menu. This panel may be used to create a DB2 role in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE ROLE statement in the SELCOPY/i foreground.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL CREATE ROLE generated by the specified input field parameters.
Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing SQL command CREATE ROLE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL CREATE ROLE statement syntax generated for the selected input field values.

Help

Open the general help for the Create Role panel.

Field Entries

Role:

Identifies the DB2 role name to be created at the current server location. The role name must not match the name of an existing role.

Name>

The SQL identifier name of the new role.
The role name must not be SYSADM, SYSCTRL or PUBLIC and has a maximum length of 128 characters.
This parameter field is mandatory and corresponds to SQL CREATE ROLE parameter *role-name*.

Drop DB2 Objects

The Drop DB2 Object panel (ZZS2D000) is an interactive panel window, opened on selection of option 6. in the DB2 Primary options menu.

This panel allows the user to select the type of object to be dropped by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Note that successful drop of some DB2 objects is dependent upon the version of DB2 used by the connected DB2 system to which this panel applies. An SQL error message will occur if an SQL DROP parameter field is used which is unsupported by the DB2 version. Please refer to the relevant edition of the *"z/OS SQL Reference"*.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Help

Open the general help for the Drop DB2 Objects option menu panel.

Options

1. Storage group
2. Database
3. Table space
4. Table

9. Type
10. Function
11. Stored procedure
12. Trigger

5. View
6. Alias
7. Index
8. Synonym
13. Global temporary table
14. Sequence
15. Role
16. Trusted context

Drop Storage Group

The Drop Storage Group panel (ZZS2DSG0) is an interactive panel window, opened on selection of option 1. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 storage group in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP STOGROUP generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP STOGROUP syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP STOGROUP statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop Storage Group panel.

Field Entries

`Storage Group Name>`

Identifies the name of the DB2 storage group to be dropped from the current server but not one that is in use by a table space or index space. Maximum length of a storage group name is 128 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter STOGROUP *stogroup-name*.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Database

The Drop Database panel (ZZS2DDB0) is an interactive panel window, opened on selection of option 2. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 database in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`
Executes the SQL DROP DATABASE generated by the specified input field parameters.

`Command`
Opens a text edit view for a temporary data set containing SQL command DROP DATABASE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`
Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP DATABASE statement syntax generated for the selected input field values.

`Help`
Open the general help for the Drop Database panel.

Field Entries

`Database Name>`
Identifies the name of the DB2 database to be dropped from the current server. Dropping a database will also drop all of its table spaces, tables, index spaces and indexes. Maximum length of a database name is 8 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter DATABASE *database-name*.

`Enter>`
Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Table Space

The Drop Table Space panel (ZZS2DTS0) is an interactive panel window, opened on selection of option 3. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 table space in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP TABLESPACE generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP TABLESPACE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP TABLESPACE statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop Table Space panel.

Field Entries

`Table Space:`

Fields that together identify the DB2 table space name to be dropped from the current server location. Dropping a table space will also drop all of its tables.

`Table Space Name>`

The table space name of the table space to be dropped.
The table space name must not identify a table space implicitlty created for an XML column, and has a maximum length of 8 characters.

`Database Name>`

The name of the data base in which the table space is defined. If not specified, database name DSNDB04 is used.
The database name has a maximum length of 8 characters.

This parameter field corresponds to SQL DROP parameter TABLESPACE *database-name.table-space-name*.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Table

The Drop Table panel (ZZS2DTA0) is an interactive panel window, opened on selection of option 4. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 table in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP TABLE generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP TABLE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP TABLE statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Table panel.

Field Entries

Table:

Fields that together identify the DB2 table name to be dropped from the current server location. Dropping a table will also drop all aliases, synonyms, views indexes and privileges on that table; all referential constraints in which the table is a parent or dependent and, if implicitly created, the table space containing the table.

Owner>

The table owner (schema) of the table to be dropped.
The table owner name has a maximum length of 128 characters.

Name>

The SQL identifier name of the table to be dropped.
The table name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter TABLE *table-name*.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop View

The Drop View panel (ZZS2DVI0) is an interactive panel window, opened on selection of option 5. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 view in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP VIEW generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP VIEW syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP VIEW statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop View panel.

Field Entries

`View:`

Fields that together identify the DB2 view name to be dropped from the current server location. Dropping a view will also drop all synonyms, other views or materialised query tables defined on the view and privileges on the view.

`Owner>`

The view owner (schema) of the view to be dropped.
The view owner name has a maximum length of 128 characters.

`Name>`

The SQL identifier name of the view to be dropped.
The view name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter VIEW *view-name*.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Alias

The Drop Alias panel (ZZS2DAL0) is an interactive panel window, opened on selection of option 6. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 alias in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP ALIAS generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP ALIAS syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP ALIAS statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Alias panel.

Field Entries

Alias:

Fields that together identify the DB2 alias name to be dropped from the current server location.

Owner>

The alias owner (schema) of the alias to be dropped.
The alias owner name has a maximum length of 128 characters.

Name>

The SQL identifier name of the alias to be dropped.
The alias name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter ALIAS *alias-name*.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Index

The Drop Index panel (ZZS2DIN0) is an interactive panel window, opened on selection of option 7. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 index in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP INDEX generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP INDEX syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP INDEX statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop Index panel.

Field Entries

`Index:`

Fields that together identify the name of the user defined DB2 index to be dropped from the current server location. Dropping an index will also drop the index space containing the index.

`Owner>`

The index owner (schema) of the index to be dropped.
The index owner name has a maximum length of 128 characters.

`Name>`

The SQL identifier name of the index to be dropped.
The index name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter INDEX *index-name*.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Synonym

The Drop Synonym panel (ZZS2DSY0) is an interactive panel window, opened on selection of option 8. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 synonym in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP SYNONYM generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP SYNONYM syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP SYNONYM statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Synonym panel.

Field Entries

Synonym:

Specifies the DB2 synonym name, for which the user is the owner, to be dropped from the current server location.

Name>

The SQL identifier name of the synonym to be dropped.
The synonym name has a maximum length of 128 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter SYNONYM *synonym-name*.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Type

The Drop Type panel (ZZS2DTY0) is an interactive panel window, opened on selection of option 9. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined distinct type in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP TYPE generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP TYPE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP TYPE statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Type panel.

Field Entries

Distinct Type:

Fields that together identify the name of the user defined DB2 distinct type to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a distinct type if it is used by dependent operations, is default and cannot be overridden by this panel.

Schema>

The schema of the distinct type to be dropped.
The distinct type schema name has a maximum length of 128 characters.

Name>

The SQL identifier name of the distinct type to be dropped.
The distinct type name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter TYPE *distinct-type-name*.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Function

The Drop Function panel (ZZS2DFU0) is an interactive panel window, opened on selection of option 10. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined function in the current DB2 subsystem by its function name. Dropping a function by its function signature or by its specific name is not supported.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP FUNCTION generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP FUNCTION syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP FUNCTION statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Function panel.

Field Entries

Function:

Fields that together identify the name of the DB2 function which was created using a CREATE FUNCTION statement and is now to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a function if it is used by dependent operations, is default and cannot be overridden by this panel.

Schema>

The schema of the function to be dropped.
The function schema name has a maximum length of 128 characters.

Name>

The SQL identifier name of the function to be dropped.
The function name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter FUNCTION *function-name* RESTRICT.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Stored Procedure

The Drop Stored Procedure panel (ZZS2DSP0) is an interactive panel window, opened on selection of option 11. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 stored procedure in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP PROCEDURE generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP PROCEDURE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP PROCEDURE statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Stored Procedure panel.

Field Entries

Stored Procedure:

Fields that together identify the name of the DB2 stored procedure which was created using a CREATE PROCEDURE statement and is now to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a stored procedure if it is used by dependent operations, is default and cannot be overridden by this panel.

Schema>

The schema of the stored procedure to be dropped.
The stored procedure schema name has a maximum length of 128 characters.

Name>

The SQL identifier name of the stored procedure to be dropped.
The stored procedure name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter PROCEDURE *procedure-name* RESTRICT.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Trigger

The Drop Trigger panel (ZZS2DTR0) is an interactive panel window, opened on selection of option 12. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined trigger in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run

Executes the SQL DROP TRIGGER generated by the specified input field parameters.

Command

Opens a text edit view for a temporary data set containing SQL command DROP TRIGGER syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

JCL

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP TRIGGER statement syntax generated for the selected input field values.

Help

Open the general help for the Drop Trigger panel.

Field Entries

Trigger:

Fields that together identify the name of the DB2 trigger to be dropped from the current server location.

Schema>

The schema of the trigger to be dropped.
The trigger schema name has a maximum length of 128 characters.

Name>

The SQL identifier name of the trigger to be dropped.
The trigger name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter TRIGGER *trigger-name*.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Global Temporary Table

The Drop Global Temporary Table panel is the same panel used for Drop Table (ZZS2DTA0) and is opened on selection of option 13. in the Drop DB2 Object options menu.

Drop Sequence

The Drop Sequence panel (ZZS2DSQ0) is an interactive panel window, opened on selection of option 14. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined sequence in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP SEQUENCE generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP SEQUENCE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP SEQUENCE statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop Sequence panel.

Field Entries

`Sequence:`

Fields that together identify the name of the DB2 sequence to be dropped from the current server location.

`Schema>`

The schema of the sequence to be dropped.
The sequence schema name has a maximum length of 128 characters.

`Name>`

The SQL identifier name of the sequence to be dropped.
The sequence name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter SEQUENCE *sequence-name*.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Role

The Drop Role panel (ZZS2DRO0) is an interactive panel window, opened on selection of option 15. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 role in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP ROLE generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP ROLE syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP ROLE statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop Role panel.

Field Entries

`Role:`

Specifies the DB2 role name to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a role if it is used by dependent operations, is default and cannot be overridden by this panel.

`Name>`

The SQL identifier name of the role to be dropped.
The role name has a maximum length of 128 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter ROLE *role-name* RESTRICT.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Trusted context

The Drop Trusted context panel (ZZS2DTC0) is an interactive panel window, opened on selection of option 16. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 trusted context profile in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the Enter> field.

Menu Bar Items

`File`

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`

Executes the SQL DROP TRUSTED CONTEXT generated by the specified input field parameters.

`Command`

Opens a text edit view for a temporary data set containing SQL command DROP TRUSTED CONTEXT syntax generated for the selected input field values. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`

Opens a text edit view for a temporary data set containing JCL that runs program DSNTIAD (via IKJEFT01) to execute the SQL DROP TRUSTED CONTEXT statement syntax generated for the selected input field values.

`Help`

Open the general help for the Drop Trusted context panel.

Field Entries

`Trusted context:`

Specifies the DB2 trusted context name to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a trusted context if it is used by dependent operations, is default and cannot be overridden by this panel.

`Name>`

The SQL identifier name of the trusted context to be dropped.
The trusted context name has a maximum length of 127 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter TRUSTED CONTEXT *context-name*.

`Enter>`

Specifies the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◊ **Nop**
Indicates that No operation is to be performed.

◊ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◊ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◊ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

List DB2 Objects

The List DB2 Object panel (ZZS2L000) is an interactive panel window, opened on selection of option 7. in the DB2 Primary options menu or option 12. 'DB2' from the List Menu.

This panel allows the user to select the type of object to be listed by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Menu Bar Items

File
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Help
Open the general help for the List DB2 Objects option menu panel.

Options

1. Storage groups
2. Databases
3. Table spaces
4. Tables
5. Views
6. Aliases
7. Indexes
8. Synonyms
9. Types
10. Triggers
11. Global temporary tables
12. Sequences
13. Roles
14. Trusted contexts
15. Columns
16. Volumes
17. Table space parts

List Storage groups

The List Storage groups panel (ZZS2LSTG) may be used to list storage groups defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 1. Storage groups from the List DB2 Object options menu.
- Execute the command **LDSTG** with or without parameters from the command line of any window.
- Execute the prefix command "**SG**" against an entry in the List Database panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSSTOGROUP. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

Name>
Used to specify a filter on storage group name.
A storage group name has a maximum length of 128 characters.

Creator>
Used to specify a filter on storage group creator (owner) authorisation ID.
A storage group creator ID has a maximum length of 128 characters.

VCatName>
Used to specify a filter on ICF catalog name/alias associated with the storage group.
The ICF catalog name/alias has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command DB.
D	Drop the storage group.
DB	List Databases in the storage group.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.

>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Databases

The List Databases panel (ZZS2LDB0) may be used to list databases defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 2. Databases from the List DB2 Object options menu.
- Execute the command **LDDB** with or without parameters from the command line of any window.
- Execute the prefix command "**DB**" against an entry in the List Storage Groups or List Table Spaces panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSDATABASE. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

Name>
Used to specify a filter on database name.
A database name has a maximum length of 8 characters.

Creator>
Used to specify a filter on database creator (owner) authorisation ID.
A database creator ID has a maximum length of 128 characters.

StGroup>
Used to specify a filter on the default storage group associated with the database.
The storage group name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command T.
D	Drop the database.
SG	List Storage Group to which the entry belongs.
T	List Tables in the database entry.
TS	List Table Spaces in the database entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Table spaces

The List Table spaces panel (ZZS2LTS0) may be used to list table spaces defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 3. Table spaces from the List DB2 Object options menu.
- Execute the command **LDTSP** with or without parameters from the command line of any window.
- Execute the prefix command "**TS**" against an entry in the List Database panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLESPACE. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Name>`
Used to specify a filter on table space name.
A table space name has a maximum length of 8 characters.

`Creator>`
Used to specify a filter on table space creator (owner) authorisation ID.
A table space creator ID has a maximum length of 128 characters.

`DBName>`
Used to specify a filter on the Database name to which the table space belongs.
The database name has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command T.
D	Drop the table space.
DB	List database to which the entry belongs.
T	List Tables in the table space entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Tables

The List Tables panel (ZZS2LTAB) may be used to list tables defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 4. Tables from the List DB2 Object options menu.
- Execute the command **LDTAB** with or without parameters from the command line of any window.
- Execute the prefix command "**T**" against an entry in the List Databases, List Table Spaces or List Tablesapce partitions panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Name>`
Used to specify a filter on table name.
A table name has a maximum length of 128 characters.

`Creator>`
Used to specify a filter on table creator (owner) authorisation ID.
A table creator ID has a maximum length of 128 characters.

DBName>

Used to specify a filter on the database name to which the table belongs.
The database name has a maximum length of 8 characters.

TSName>

Used to specify a filter on the table space name in which the table is defined.
The table space name has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
A	Create an alias for the table entry.
AL	List aliases for the table entry.
B	Open the DB2 Browse panel to browse contents of the table entry.
BI	Browse the contents of the table entry.
CL	List columns belonging to the table entry.
D	Drop the table.
E	Open the DB2 Edit panel to edit the contents of the table entry.
EI	Edit the contents of the table entry.
S	Create a synonym for the table entry.
SC	Create a SELCOPY batch job using the DB2 table as input.
SL	List synonyms for the table entry.
T	Create a trigger for the table entry.
TL	List triggers for the table entry.
V	Create a view using the table entry.
VL	List views using the table entry.
XL	List indexes using the table entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Views

The List Views panel (ZZS2LVI0) may be used to list views defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 5. Views from the List DB2 Object options menu.
- Execute the command **LDVIEW** with or without parameters from the command line of any window.
- Execute the prefix command "**VL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog table.

List columns are those defined in the DB2 catalog view SYSIBM.SYSVIEWS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

Owner>

Used to specify a filter on view creator (owner) authorisation ID.
A view creator ID has a maximum length of 128 characters.

Name>

Used to specify a filter on view name.
A view name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the view entry.
CL	List columns belonging to the view entry.
D	Drop the view.
E	Open the DB2 Edit panel to edit the contents of the view entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Aliases

The List Aliases panel (ZZS2LALI) may be used to list aliases defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 6. Aliases from the List DB2 Object options menu.
- Execute the command **LDALI** with or without parameters from the command line of any window.
- Execute the prefix command "**AL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Owner>`

Used to specify a filter on alias creator (owner) authorisation ID.
An alias creator ID has a maximum length of 128 characters.

`Name>`

Used to specify a filter on alias name.
An alias name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the table to which the alias entry applies.
CL	List columns belonging to the table to which the alias entry applies.
D	Drop the alias.
E	Open the DB2 Edit panel to edit the contents of the table to which the alias entry applies.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Indexes

The List Indexes panel (ZZS2LINX) may be used to list indexes defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 7. Indexes from the List DB2 Object options menu.
- Execute the command **LDINDEX** with or without parameters from the command line of any window.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSINDEXES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Index Creator>`
Used to specify a filter on index creator (owner) authorisation ID.
An index creator ID has a maximum length of 128 characters.

`(Index) Name>`
Used to specify a filter on index name.
An index name has a maximum length of 128 characters.

`Table Creator>`
Used to specify a filter on table creator (owner) authorisation ID for the table on which the index is defined.
A table creator ID has a maximum length of 128 characters.

`(Table) Name>`
Used to specify a filter on table name for the table on which the index is defined.
A table name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the table to which the index entry applies.
CL	List columns belonging to the table to which the index entry applies.
D	Drop the index.
E	Open the DB2 Edit panel to edit the contents of the table to which the index entry applies.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Synonyms

The List Synonyms panel (ZZS2LSYN) may be used to list synonyms defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 8. Synonyms from the List DB2 Object options menu.
- Execute the command **LDSYN** with or without parameters from the command line of any window.
- Execute the prefix command "**SL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSSYNONYMS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Creator>`

Used to specify a filter on synonym creator (owner) authorisation ID.
A synonym creator ID has a maximum length of 128 characters.

`Name>`

Used to specify a filter on synonym name.
A synonym name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the table to which the synonym entry applies.
CL	List columns belonging to the table to which the synonym entry applies.
D	Drop the synonym.
E	Open the DB2 Edit panel to edit the contents of the table to which the synonym entry applies.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Distinct Types

The List Distinct Types panel (ZZS2LTYP) may be used to list distinct types defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 9. Types from the List DB2 Object options menu.
- Execute the command **LDTYP** with or without parameters from the command line of any window.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSDATATYPES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Owner>`

Used to specify a filter on distinct type creator (owner) authorisation ID.
A distinct type creator ID has a maximum length of 128 characters.

`Name>`

Used to specify a filter on distinct type name.
A distinct type name has a maximum length of 128 characters.

Prefix Line Commands

List of Distinct Types does not support any prefix commands, therefore no prefix area is included in the display.

List Triggers

The List Triggers panel (ZZS2LTRG) may be used to list triggers defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 10. Triggers from the List DB2 Object options menu.
- Execute the prefix command "**TL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSTRIGGERS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Owner>`

Used to specify a filter on trigger creator (owner) authorisation ID.
A trigger creator ID has a maximum length of 128 characters.

`Name>`

Used to specify a filter on trigger name.
A trigger name has a maximum length of 128 characters.

Prefix Line Commands

List of Triggers does not support any prefix commands, therefore no prefix area is included in the display.

List Global Temporary Tables

The List Global Temporary Tables panel (ZZS2LGTT) may be used to list tables defined as being temporary in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 11. Tables from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Owner>`

Used to specify a filter on global temporary table creator (owner) authorisation ID.
A table creator ID has a maximum length of 128 characters.

`Name>`

Used to specify a filter on the global temporary table name.
A table name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
CL	List columns in the global temporary table.
D	Drop the global temporary table.
B	Browse the tables contents.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Sequences

The List Sequences panel (ZZS2LSEQ) may be used to list sequences defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 12. Sequences from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSSEQUENCES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

Owner>

Used to specify a filter on sequence creator (owner) authorisation ID.
A sequence creator ID has a maximum length of 128 characters.

Name>

Used to specify a filter on sequence name.
A sequence name has a maximum length of 128 characters.

Prefix Line Commands

List of Sequences does not support any prefix commands, therefore no prefix area is included in the display.

List Roles

The List Roles panel (ZZS2LROL) may be used to list roles defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 13. Roles from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSROLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Definer>`
Used to specify a filter on the authorisation ID (or role) that defined the role name.
A role authorisation ID or role name has a maximum length of 128 characters.

`Name>`
Used to specify a filter on role name.
A role name has a maximum length of 128 characters.

Prefix Line Commands

List of Roles does not support any prefix commands, therefore no prefix area is included in the display.

List Trusted Contexts

The List Trusted Context panel (ZZS2LTRC) may be used to list trusted context defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 14. Trusted Contexts from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSCONTEXT. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Definer>`
Used to specify a filter on the authorisation ID (or role) that defined the trusted context.
An authorisation ID or role name has a maximum length of 128 characters.

`Name>`
Used to specify a filter on the trusted context name.
A trusted context name has a maximum length of 128 characters.

Prefix Line Commands

List of Trusted Contexts does not support any prefix commands, therefore no prefix area is included in the display.

List Columns

The List Columns panel (ZZS2LCOL) may be used to list columns defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 15. Columns from the List DB2 Object options menu.
- Execute the command **LDCOL** with or without parameters from the command line of any window.
- Execute the prefix command "**CL**" against an entry in the List Tables, List Views, List Aliases, List Indexes, List Synonyms or List Global Temporary Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog table.

List columns are those defined in the DB2 catalog column SYSIBM.SYSCOLUMNS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

Table Creator>
Used to specify a filter on the creator (owner) of the table or view that contains the column.
A table or view creator ID has a maximum length of 128 characters.

Table Name>
Used to specify a filter on the name of the table or view that contains the column.
A table or view name has a maximum length of 128 characters.

Name>
Used to specify a filter on column name.
A column name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the column entry.
BT	Browse the contents of the table to which the column entry belongs.
E	Open the DB2 Edit panel to edit the contents of the table to which the column entry belongs.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

List Volumes

The List Volumes panel (ZZS2LVOL) may be used to list volumes defined to storage groups in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 16. Volumes from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSVOLUMES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

Owner>
Used to specify a filter on storage group creator (owner) authorisation ID to which the volume is defined.
A storage group authorisation ID a maximum length of 128 characters.

Name>
Used to specify a filter on the storage group to which the volume is defined.
A storage group name has a maximum length of 128 characters.

Prefix Line Commands

List of Volumes does not support any prefix commands, therefore no prefix area is included in the display.

List Table space parts

The List Table space parts panel (ZZS2LTS1) may be used to list table space partitions defined in the current DB2 subsystem.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 17. Table space parts from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLEPART. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

TSName>
Used to specify a filter on the table space name.
A table space name has a maximum length of 8 characters.

DBName>
Used to specify a filter on the Database name to which the table space belongs.
The database name has a maximum length of 8 characters.

VCAT Name>
Used to specify a filter on ICF catalog name/alias used for table space partition free space allocation.
The ICF catalog name/alias has a maximum length of 8 characters.

Stor Name>
Used to specify a filter on the storage group used for table space allocation.
The storage group name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command T.
D	Drop the table space partition.
DB	List database to which the entry belongs.
T	List Tables in the table space partition entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to PF4 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF2 by default.

Audit Trail Functions

The List DB2 Object panel (ZZS2AUD0) is an interactive panel window, opened on selection of option 8. in the DB2 Primary options menu.

This panel includes options relating to management and display of SELCOPY/i DB2 audit log data sets.

SELCOPY/i DB2 supports the generation of audit log data sets that record all SQL activity that has occurred during a DB2 connection. Unless logging is deactivated, a single audit log data set is automatically allocated for each DB2 subsystem connection. All SQL statements executed using this subsystem connection will be logged in the audit file.

Except for DB2 edit, which maintains a separate log file for each table edited, a DB2 connection and log file will be generated once for each execution of the DB2 primary option menu panel.

Users may choose to maintain a log data set via a flag in the DB2 primary option menu panel or, for table edit, a flag in the Edit Object panel or EDIT line command. Data set options, used by SELCOPY/i to allocate a new log data set, may be customised via the Audit Log Dataset Options panel. Other Audit panels provide facilities for listing and printing Audit data sets.

Menu Bar Items

`File`
: The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`
: Open the general help for the Audit Trail Functions option menu panel.

Options

1. Audit Log Dataset Options
2. Print Audit Report
3. List Audit Datasets

Audit Log Dataset Options

The Audit Log Allocation panel (ZZS2AUDS) is an interactive panel window, opened on selection of option 1. in the DB2 Audit Trail Functions options menu.

This panel allows the user to configure data set options that are subsequently used by SELCOPY/i when allocating new DB2 audit log data sets for the user. Note that SELCOPY/i DB2 logs are RECFM=VB physical sequential data sets.

Options in this panel should be customised so that log data sets comply with your system standards.

Panel Input Fields

`High Level Qualifier:`
: Specifies the data set name high level qualifier prefix to be used. Mutually exclusive options are as follow:

  ◊ **Use TSO Prefix**
  Use the TSO PREFIX value as set by the user's profile.

  ◊ **Use User ID**
  Use the user's TSO (or SELCOPY/i VTAM) logon id.

  ◊ **Use Specified HLQ**
  Use the HLQ specified by field entry HLQ>

`HLQ>`
: Applicable only if option Use Specified HLQ is selected. this field names the SDE edit profile macro to be executed when the table is edited.

`Device Type:`
: The device or type of device on which the log data set should be allocated.

  `Unit>`
  : Specifies the UNIT device number, device type or esoteric group name.
  Note that no UNIT parameter is required if the log data set is SMS managed. Specify a STORCLAS or let an automated class selection (ACS) routine select a storage class for the data set.

`Allocation Unit:`
: Identifies the SPACE unit of allocation. Mutually exclusive options are as follow:

  ◊ **Cylinders**
  Requests that the space be allocated in DASD cylinders.

  ◊ **Tracks**
  Requests that the space be allocated in DASD tracks.

- ◊ **Blocks**
  Requests that the space be allocated in blocks.

`Allocation Size:`
Fields relating to the number of SPACE allocation units to allocate.

`Primary>`
Primary quantity of allocated units.

`Secondary>`
Secondary quantity of allocated units.

`SMS Classes:`
Fields relating to SMS data set management.

`Data Class>`
SMS Data Class to be used. Specify a Data Class if one is not automatically selected via an ACS routine.

`Storage Class>`
SMS Storage Class to be used. Specify a Storage Class if one is not automatically selected via an ACS routine.

`Management Class>`
SMS Management Class to be used. Specify a Management Class if one is not automatically selected via an ACS routine.

Print Audit Report

The Audit Log Report panel (ZZS2AUDP) is an interactive panel window, opened on selection of option 2. in the DB2 Audit Trail Functions options menu.

This panel allows the user to invoke the AUDPRINT command to display print output of a selected SELCOPY/i DB2 audit log. Printing an audit log, first processes the formatted records of the audit log file to generate a printable report.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to display the report in a temporary text edit view.

Menu Bar Items

`File`
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Run`
Execute AUDPRINT in the SELCOPY/i foreground to formats the audit log records and display the printed report in a temporary text edit view.
Hitting <Enter> will perform the same action.

`Command`
Opens a text edit view for a temporary data set containing the AUDPRINT command syntax generated for the selected DB2 audit log data set. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <PF4>. (i.e. using the CMDTEXT facility.)

`JCL`
Opens a text edit view for a temporary data set containing JCL that runs program SELCOPY in batch to format the log records and write the generated report to SYSPRINT.

`Help`
Open the general help for the DB2 Audit Log Report panel.

Panel Input Fields

`Audit DSN:`

Specifies the DSN of the SELCOPY/i DB2 audit log file to print.

The format of a log file DSN is *prefix*.ZZSX.*ssn*.Dyyyyddd.Thhmmsss.AUD and, by default, this field entry displays the DSN of the last log file created by SELCOPY/i for the user.

List Audit Datasets

The List Audit Datasets window is simply a List Catalog Entries window with the the default DB2 audit log DSN mask passed to the **Entry>** field so that only the audit log entries are displayed.

Use the "AP" prefix command to open the Print Audit Report panel to display a printable report of any log file entry in this list.

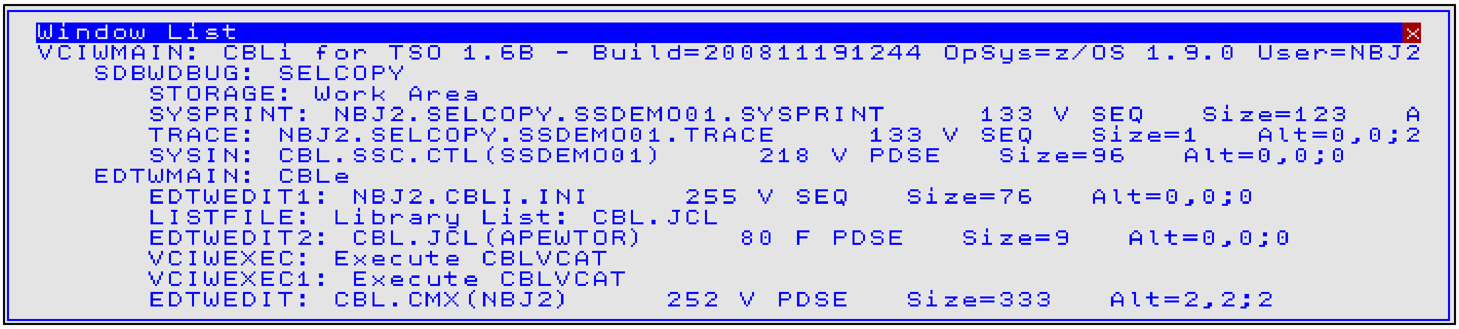
Window List (=W)

The Window List window may be opened via the following:

- Select option W. 'Window' from the Primary Option Menu panel menu bar.
- Select 'Window List' from the Primary Option Menu panel.
- Select 'All Windows' from the Window menu in the CBLe main window menu bar.
- Enter the command WINDOWLIST on the command line of any window.

The Window List window displays all open windows and their associated window names and allows the user to place focus on a specific window by selecting it from the list.

The hierarchy of parent/child windows is illustrated by indentation of the entries in the list.



```
Window List
VCIWMAIN: CBLi for TSO 1.6B - Build=200811191244 OpSys=z/OS 1.9.0 User=NBJ2
   SDBWDBUG: SELCOPY
      STORAGE: Work Area
      SYSPRINT: NBJ2.SELCOPY.SSDEMO01.SYSPRINT     133 V SEQ   Size=123   A
      TRACE: NBJ2.SELCOPY.SSDEMO01.TRACE     133 V SEQ   Size=1   Alt=0,0;2
      SYSIN: CBL.SSC.CTL(SSDEMO01)     218 V PDSE   Size=96   Alt=0,0;0
   EDTWMAIN: CBLe
      EDTWEDIT1: NBJ2.CBLI.INI     255 V SEQ   Size=76   Alt=0,0;0
      LISTFILE: Library List: CBL.JCL
      EDTWEDIT2: CBL.JCL(APEWTOR)     80 F PDSE   Size=9   Alt=0,0;0
      VCIWEXEC: Execute CBLVCAT
      VCIWEXEC1: Execute CBLVCAT
      EDTWEDIT: CBL.CMX(NBJ2)     252 V PDSE   Size=333   Alt=2,2;2
```

Figure 128. Window List window.

SELCOPY/i Command Reference

SELCOPY/i commands may be issued from a command line at the Command> prompt.
All SELCOPY/i main window menu bar commands have a command line equivalent.

Command	Description
ALIAS	Open the Create ALIAS dialog window (includes support for PDSE Load libraries).
AMS	Open the IDCAMS Command window. An AMS/IDCAMS command may be passed as a parameter.
AMSDIALOG	Open the Execute IDCAMS dialog panel.
APE	Open the SELCOPY/i Module List.
AUDPRINT	Display formatted DB2 Audit log output.
BACKWARD	Scroll the display backwards by one page.
BOTTOM	Display the last lines of data.
BROWSE	Open the CBLe text editor to edit a file read-only.
CALENDAR	Open the calendar window.
CALC	Open the calculator window.
CBLI	Pass a command directly to the SELCOPY/i command processor.
CBLICANCEL	Exit the SELCOPY/i session without opening the quit session confirmation pop-up mesage.
CBLNAME	Open a storage window containing the loaded CBLNAME module.
CFOUT	Display SDE formatted Compare File report output.
CLOSE	Close a window.
CMDTEXT	Pick up a command from the text displayed in a window at the cursor position and either execute it or place it on the command line for editing before execution.
COMMANDLINE	Set the command line attributes.
COMPFILE	Execute the Compare File Utility.
COMPLIB	Execute the Compare Library Utility.
CURSORSELECT	Perform the default operation based on the cursor position.
DB2	Open the SELCOPY/i DB2 panels.
DCMD	Execute the DB2 Command execution panel.
DOWN	Scroll the window display downwards.
DRAGBORDERMINUS	Drag the window's border closer to the top left corner of the display.
DRAGBORDERPLUS	Drag the window's border away from the top left corner of the display.
DSQL	Execute the SQL Statement execution panel.
EDIT	Open the CBLe text editor to edit a file.
EO	Display output queue listing.
ERASE	Erase a file.
FAV	Open the Favourites Panel.
FCOPY	Execute the File Copy Utility.
FORWARD	Scroll the display forwards by one page.
FS	Open the File Search window.
FSU	Execute the File Search/Update/Copy/Remap Utility.
FSUEND	Close an FSU report output display.
FSUOUT	Display SDE formatted FSU report output.
FSUUNDO	Execute the FSU Undo Updated records utility.
HELP	Open the help top level window.
HOME	Place focus on the CBLe edit view containing the user's command centre file.
IEBCOPYDIALOG	Open the IEBCOPY Dialog window.
ISPF	Toggle between TSO and ISPF 3270 I/O.
ISPFUTIL	Start ISPF Utilities Panel.
KEYS	Set a function key or open the function key dialog.
LA	Open the list allocated files window.

LAS	List VSAM Associated objects.
LC	Open the cataloged files list window. A fileid mask may be passed as a parameter. For VSE a catalog DLBL is required.
LD	Open the dataset details list window. For MVS a partial dataset name may be passed as a parameter. For VSE not yet implemented. For CMS a file name, type and mode pattern may be passed as a parameter.
LEFT	Scroll the window display to the left.
LJQ	Open the Job Enqueues list window. A job name may be passed as a parameter. Not implemented for VSE.
LL	Open the library members list window. For MVS a PDS name may be passed as a parameter. For VSE a LIBR library, sublibrary and member name and type pattern may be passed as a parameter.
LLX	Execute the Locat Library Members utility
LP	Open the HFS Paths list window. An absolute or relative HFS path may be passed as a parameter.
LQ	Open the Enqueues list window. An enqueue major name and resource name may be passed as a parameter. Not implemented for VSE.
LV	Open the VTOC files list window. A volume name may be passed as a parameter.
LVOL	Open the DASD volumes list window. A volume name pattern may be passed as a parameter.
LVR	Open the CBLVCAT Raw list window. A CBLVCAT command string may be passed as a parameter.
LX	Open the VTOC extents list window. A volume name may be passed as a parameter. Not yet implemented for VSE.
MAXIMISE	Maximise a window.
MDINEXT	Place focus on the next MDI child window.
MDIPREV	Place focus on the previous MDI child window.
MINIMISE	Minimise a window.
MOVEWINDOW	Move a window.
NEXTMAINWINDOW	Sets the focus window to the next main window within the SELCOPY/i desktop.
NEXTWINDOW	Sets the focus window to the next window in the ring of all windows.
POWER	Open the POWER Command Output window.
PREVMAINWINDOW	Sets the focus window to the previous main window within the SELCOPY/i desktop.
PREVWINDOW	Sets the focus window to the previous window in the ring of all windows.
QUIT	Exit and close the current window.
RENAME	Rename a file.
RESTORE	Restore a window.
RETRIEVE	Retrieve previously executed commands.
RIGHT	Scroll the window display to the right.
SDATA	Direct a command to the Structured Data Environment (SDE).
SDE	Same as SDATA but opens SDE dialog window if no parameters.
SDSF	Start the SDSF application.
SELCOPY	Start the SELCOPY Debug window.
SETCOLOUR	Remap the appearence of a colour and its associated highlighting.
SETFOCUS	Sets the focus window to a named window in the ring of all windows.
SHOWPOPUPMENU	For storage windows, display the options pop-up menu.
SHOWWATTR	Show Window Attributes.
SIZEWINDOW	Resize a window.
SQL	Open the Dynamic SQL window.
SVC	Display the status of the CBLVCAT Interactive SVC.
SYSAPF	Open the APF List window. (MVS only)
SYSCOMMAND	Pass a command directly to the local TSO or CMS command processor.
SYSI	Open the system information window. Not yet implemented for VSE.
SYSLL	Open the Link List window. (MVS only)
SYSLPA	Open the LPA Modules window. (MVS only)

SYSMENU	Open a window's System Menu.
SYSPGM	Open the Loaded Programs Menu.
SYSSTOR	Open the Storage Statistics window.
SYSTASK	Open the Task List window. (MVS only)
TASK	Execute a program as a sub-task of SELCOPY/i.
TOP	Display the first lines of data.
UP	Scroll the window display upwards.
VCAT	Open the Execute CBLVCAT window. A CBLVCAT command string may be passed as a parameter.
VIEW	Open the CBLe text editor to edit a file read-only.
VOLSTATS	Open the Volume statistics window.
WINDOWLIST	Open the window list window. This lists all currently open windows.
ZZSIDOWNAMES	Hide or display the window names in the title bar.

ALIAS

Syntax:

```
               +-----------------------+
               |                       |
               V                       |
>>--+- ALIAS --+-----------------------+-- library(member) -- alias --------->< 
               |                       |
               +-- -AMODE -+-- 24 --+--+
               |           |        |  |
               |           +-- 31 --+  |
               |           |        |  |
               |           +-- ANY -+  |
               |                       |
               +-- -DLG ---------------+
               |                       |
               +-- -ENTRY entry_name --+
```

Description:

Use the ALIAS command to create a new PDS/PDSE library member alias, or open "Create ALIAS" dialog window.

Note that aliases for load-library members are created using the binder to relink the module in being aliased. This will result in an update to the module's **TTR**.

Use of the **-DLG** parameter invokes the Create ALIAS dialog window, which is also available through the **'A'** prefix-command in a Library List window.

Parameters:

library(member)
: The PDS/PDSE library member be be aliased.

alias
: The alias name to be added

-AMODE
: For a load-library member, used to specify the Addressing Mode for the new aliased entry-point. Valid arguments are 24, 31 and ANY.

-DLG
: Invoke the "Create Alias" dialog window.

-ENTRY *entry_name*
: For a load-library member, used to specify the symbolic name of the entry-point address to be used.

Examples:

```
< alias JGE.CBLINST.D070919.EXE.COPY(I141) CBLI
< alias JGE.CBLINST.D070919.EXE.COPY(I141) CBLAVTAM -entry VCIAVTAM
< alias JGE.CBLINST.D070919.EXE.COPY(I141) CBLIVTAM -entry VCIRVTAM
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV31 -amode 31
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV24 -amode 24
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV24 -amode 24 -dlg
<  alias JGE.CBLINST.D070919.EXE.COPY(S202) SELCOPY
```

AMS

Syntax:

```
>>--- AMS ---+-----------------+---------------------------------------------><
             |                 |
             +-- AMS_Command --+
```

Description:

Use the AMS command to open the IDCAMS Command window and optionally execute an AMS/IDCAMS command.

The IDCAMS Command window displays the SYSPRINT/SYSLST output generated on execution of the IDCAMS command.

Parameters:

AMS_Command
AMS/IDCAMS command to be executed.

Examples:

```
AMS  LISTCAT  ENTRY(CBL.MCAT.CMP.P)  ALL
```

AMSDIALOG

Syntax:

```
                   +-- Ksds ---------+
                   |                 |
>>--- AMSDialog ---+-----------------+---------------------------------------><
                   |                 |
                   +-- Esds ---------+
                   |                 |
                   +-- Rrds ---------+
                   |                 |
                   +-- Lds ----------+
                   |                 |
                   +-- ALias --------+
```

Description:

Use the AMSDIALOG command to open the appropriate Define VSAM dialog window or Define Catalog ALIAS dialog window.

Parameters:

KSDS
ESDS
RRDS
LDS
ALIAS
Specifies the type entry of Define dialog window to open.
Default is Define VSAM KSDS.

Examples:

`AMSD  ESDS`
Open the Define VSAM ESDS dialog window.

`AMSD  ALIAS`
Open the Define Catalog ALIAS dialog window.

APE

Syntax:

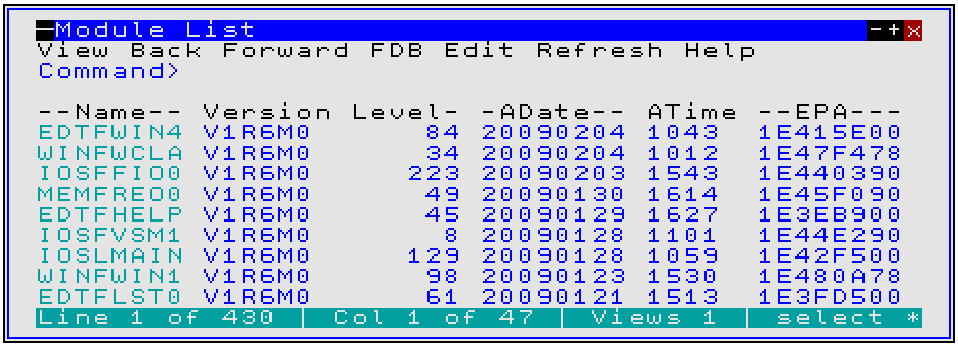
```
>>---- APE -----------------------------------------------------------------><
```

Description:

Use the APE (Assembler Programming Environment) command to display the Module List window containing the current status of all modules that comprise SELCOPY/i.

The Module List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

APE may be executed to determine the level of a currently installed module.



```
--Name-- Version Level- -ADate-- ATime --EPA---
EDTFWIN4 V1R6M0      84 20090204 1043  1E415E00
WINFWCLA V1R6M0      34 20090204 1012  1E47F478
IOSFFIO0 V1R6M0     223 20090203 1543  1E440390
MEMFREO0 V1R6M0      49 20090130 1614  1E45F090
EDTFHELP V1R6M0      45 20090129 1627  1E3EB900
IOSFVSM1 V1R6M0       8 20090128 1101  1E44E290
IOSLMAIN V1R6M0     129 20090128 1059  1E42F500
WINFWIN1 V1R6M0      98 20090123 1530  1E480A78
EDTFLST0 V1R6M0      61 20090121 1513  1E3FD500
Line 1 of 430 | Col 1 of 47 | Views 1 | select *
```

Figure 129. Module List window.

AUDPRINT

Syntax:

```
>>---- AUDPrint ----+-------------------+------------------------------------><
                    |                   |
                    +-- db2_audit_dsn --+
```

Description:

Use the AUDPRINT command to format a SELCOPY/i DB2 audit log file into a readable report and display it in a CBLe text edit view.

If specified with no *db2_audit_dsn* parameter, the Print Audit Report (ZZS2AUDP) panel is opened.

Parameters:

db2_audit_dsn
: The DSN of a SELCOPY/i DB2 audit log file.

  The high level qualifier of this DSN is determined via the Audit Log Dataset Options (ZZS2AUDS) panel.

BACKWARD

Syntax:

```
>>--+- BAck -----+-----------------------------------------------------------><
    |            |
    +- BACKWARD -+
```

Description:

Use the BACKWARD command to scroll the window contents backwards by a page.

Note that BACKWARD is the same as UP CURSOR where the cursor is outside the display area.

BOTTOM

Syntax:

```
>>--+- BOTTOM----+----------------------------------------------------------><
    |            |
    +- BOT ------+
```

Description:

Use the BOTTOM command to display the bottom lines of the data in the focus window. The last line of the data becomes the last line of the display area.

Note that the SELCOPY/i BOTTOM command acts differently to the BOTTOM command available in a CBLe text edit or SDE edit document window.

BROWSE

Syntax:

```
>>-- Browse -- fileid ---| SDE BROWSE Opts |---------------------------------><
```

Description:

For **VSE** and **CMS** systems, BROWSE is a synonym for VIEW.

Use the BROWSE command to open a Structured Data Environment (SDE) BROWSE window view to browse a page of data from the specified fileid.

If the CBLe text editor main window has been stopped, BROWSE will start the CBLe main window and open an edit view for the user's HOME CMX file before opening the SDE browse window for the requested file.

Use BROWSE instead of VIEW to browse large data sets. Unlike EDIT and VIEW, BROWSE does not need to load the entire file into storage in order to display a page of records.

Parameters:

fileid

The fileid of the file to be browsed.

fileid may be the DSN of a sequential or VSAM data set, the member name (with or without the library DSN) of a PDS/PDSE member or an HFS file name. If member name is specified without a library DSN, the DSN of the library member in the current edit view is used.

SDE BROWSE Opts

See SDE BROWSE for supported parameters.

CALENDAR

Syntax:

```
>>--+- CALENDAR -+----------------------------------------------------------><
    |            |
    +- CAL ------+
```

Description:

Use the CAL command to open the Calendar Window.

The Calendar Window may also be opened via the Utilities entry of the SELCOPY/i main menu.

CALC

Syntax:

```
>>--+- CALCULATE -+--+---------------------+--------------------------------><
    |             |  |                     |
    +- CALC ------+  +-- REXX_expression --+
```

Description:

Use the CALC command to open the Calculator Window and optionally evaluate a REXX expression.

The Calculator Window may also be opened via the Utilities menu of the SELCOPY/i main window menu bar.

Parameters:

REXX_expression
: The Calculator will evaluate any valid REXX expression. This may include the result of any REXX function built in to REXX or written by the user.

Examples:

```
CALC  (1024+281) / (3*2)

CALC  c2x(bitxor('af'x,'44'x))
```

CBLI

Syntax:

```
>>-- CBLI --- command -------------------------------------------------------><
```

Description:

Execute a SELCOPY/i command.

The specified command is passed to the SELCOPY/i environment. Any windows opened are child windows of the SELCOPY/i main window, not of the current window.

Parameters:

command
: SELCOPY/i command.

Examples:

```
cbli   vcat < cbl.vvc.ctl(vvrep01)
```

Open a CBLVCAT Interactive window and execute CBLVCAT with control statements from file CBL.VVC.CTL(VVREP01).

CBLICANCEL

Syntax:

```
>>-- CBLICANcel -------------------------------------------------------------><
```

Description:

Exit and close the SELCOPY/i session without opening the quit session confirmation pop-up window.

CBLNAME

Syntax:

```
>>-- CBLNAME --------------------------------------------------------------><
```

Description:

Use the CBLNAME command to open the CBLNAME window.

The CBLNAME storage display window may also be opened via the System menu item of the SELCOPY/i main window menu bar.

CFOUT

Syntax:

```
>>---- CFOUT -------+-------------------------+------------------------------><
                    |                         |
                    +-- compfile_report_dsn --+
```

Description:

Use the CFOUT command to display a saved SELCOPY/i Compare Files utility output report in in an SDE Browse window view.

CFOUT will automatically attempt to format the report records using an SDE structure (SDO) with DSN *compfile_report_dsn***.SDO**. If this structure does not exist, CFOUT will fail with error ZZSD024E structure not found.

Parameters:

compfile_report_dsn
The DSN of a SELCOPY/i Compare Files utility output report.

The default report DSN is '*userid*.SELCOPYI.COMPFILE.REPORT' where *userid* is the user's RACF logon id.

CLOSE

Syntax:

```
>>--+- CLOSE -+---+--------------+-----------------------------------------><
    |         |   |              |
    +- CLO ---+   +- windowname -+
    |         |
    +- Quit --+
```

Description:

Use the CLOSE command to close a window.

By default this command is assigned to function key **PF3**. You can also use the close button if the window has one.

Parameters:

windowname
The window name of the window to close. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CMDTEXT

Syntax:

```
>>--+-- CMDTEXT -----+---+-----------+--------------------------------------><
    |                |   |           |
    +-- COMMANDTEXT -+   +- EDIT ----+
    |                |   |           |
    +-- CTX ---------+   +- EDITALL -+
```

Description:

CMDTEXT extracts the text at the cursor position and either executes it as a command or places it on the command line for editing and/or execution.

By default this command is assigned to function key **PF4**.

Its prime use is from within the CBLe file editor and provides the user with a facilty to store commonly used, or difficult to remember, commands as plain text within any editable file.

While it is recommended to create dedicated 'command files', using a filetype CMX, both for specific tasks (e.g. hlq.MONTHLY.REPORT.CMX) and for one for each user with various task sections (e.g. userid.CMX), commands may also be included as comment data in JCL, program source and configuration files.

The text selected for execution can span multiple lines using the continuation character, backslash ('\' - X'E0'), allowing both for very long commands and the chaining of multiple commands using the command delimiter character. (See below)

Note: The selected text may extend beyond the text visible in the window. i.e. text that would be displayed if the window is made wider or scrolled may also form part of the CMDTEXT selected text string.

Some characters within the line of command text are given special treatment:

'|' (or symbol - X'4F')
: Used to partition a single line of command text into separately executable commands and/or sections of comment data. This character marks the limits the focus command to be executed based on the cursor position when CMDTEXT is executed. A '||' prefix can be added at the start of the line of command text (but following '<' or '>') to indicate that any occurrence of the '|' character in the command text is to be treated as part of the command data.

'<' (less than - X'4C'
: If found in the first 4 characters (including leading blanks) of the line of command text, indicates that immediate execution is required. Characters in front of the '<' are ignored. This allows comment indication characters to exist and so enabling insertion of point-and-shoot commands in JCL, etc. e.g.

```
//*<sub ;OQ  %JobName%    | Submit and wait in SDSF for output.
```

'>' (greater than - X'6E'
: As above for '<' but indicating that the command is to be placed on the command-line, for potential edit and so a delayed execution. This is the default if neither '<' nor '>' is found in the first four characters.

'_' (underscore - X'6D'
: The first occurrence of this character is always removed from the command text and indicates the location at which the cursor is positioned when the command is placed on the command line for edit. Note that, if the first occurrence of '_' in the command is to be treated as part of the command data, then an extra '_' must be added before the before it. e.g.

```
>e my.jcl(Job_01)          | Change the job number then hit <Enter> to EDIT it.
```

'`' (reverse apostrophe - X'79'
: This character is treated as a null and all occurrences are removed from the command text. Its purpose is to allow alignment of alike items purely for readabilty. e.g.

```
>e my.jcl```(Job01)        | Edit DSN 'my.jcl(Job01)'.
>e my.output(Job01)        |
```

'\' (backslash - X'E0'
: If the last non-blank character on the whole line is '\', then the line of command text is continued onto the next line. e.g.

```
>alloc  reuse f(OUTDD)  new dsn('CBL.TEST.OUTPUT') \
        space(1,1) cyl unit(3390) vol('DATT0B')     \
        recfm(F,B) lrecl(80) blksize(0)

>sub my.jcl```(Job01) \
;e   my.output(Job01) \
;e                              CBL.TEST.OUTPUT
```

Note: In the above example, ';' (semi-colon) is the command delimiter.

The command text selected by CMDTEXT is identified by starting at the cursor position and searching to the left for a '<', '>', '|' character or the start of the line. This determines the left extent of the command string. The right extent of command string is then determined by starting at the cursor and searching to the right for a '|' character or the end of the line of text.

Cursor Positioning in a CBLe Text Edit View

By default, if the executed command causes the display to scroll so that the focus line is no longer visible, then the cursor will be re-positioned in column 1 of the command line.

For CBLe text edit views, the cursor will remain on the same column and line within the window display if both of the following conditions are true:

1. The command being executed is a CBLe CLI LOCATE command with a named line-target (label name) argument. Note that the LOCATE verb is default and may be omitted. e.g. `.SDE`
2. The focus line occupies one of the first 4 lines of the window display.

This allows users to place reciprocol named line-target (label name) LOCATE commands at fixed positions at the start of each logical section within a CMX (HOME) file.

e.g. 2 sections within a CMX file are marked by the line labels ".FIND" and ".RESET" respectively. The 2nd line of the ".FIND" section has a `.RESET` locate command starting at column 12. Likewise, the 2nd line of the ".RESET" section has a `.FIND` locate command starting at column 12. Navigation back and fore between the two related sections is made easy as the cursor does not move within the display when PF4 is hit on each locate command.

Parameters:

null

If the left command string delimiter is a < character then execute the command string. Otherwise place the command string on the command line.

`EDIT`

If the left command string delimiter is a < character then place the command string on the command line. Otherwise execute the command string.

`EDITALL`

Place the whole focus line (the line containing the cursor) on the command line.

Examples:

The following are examples of commands that may be entered on any SELCOPY/i command line.

In each case a window will be created. You can close these windows with the CLOSE command (by default assigned to **PF3**) or by using the close button at the right of the title bar.

Command	Description
`<vcat q cblname`	Open the CBLVCAT execution window and list the contents of CBLNAME.
`<lvol *`	Open the list DASD volumes window to list all volumes.
`<lc sys1.h`	Open the list calatog window to list all cataloged datasets whose names begin sys1.h (MVS only).
`<lc %user%`	Open the list calatog window to list all cataloged datasets whose names begin with the current userid.
`<ll sys1.help`	Open the list library window to list all members of the sys1.help library (MVS only).
`<ll prd2.*`	Open the list library window to list all sublibraries of the prd2 library (VSE only).
`<wl`	Open the window list window.
`<cal`	Open the calendar window.
`<calc x2d('1000')`	Convert X'1000' to decimal using the calculator window.

```
<ll %user%.cbli.cble; where User = %user% & LastMod => '%date%'  \
;ll %user%.cbli.cmx ; where User = %user% & LastMod => '%date%'
```

| List all library members changed by me today.

See Also:

CBLe command, EXTRACT
CBLe command, QUERY

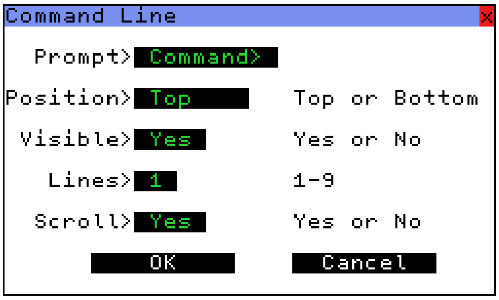
COMMANDLINE

Syntax:

```
>>--+- COMMANDLINE -+--+--------------+--+--------+--+-------+--------------->
    |               |  |              |  |        |  |       |
    +- CLN ---------+  +- windowname -+  +- SHOW -+  +- TOP -+
                                         |        |  |       |
                                         +- HIDE -+  +- BOT -+

 >--+-----------------+--+--------------+-------------------------------------><
    |                 |  |              |
    +- PROMPT=prompt -+  +- SCROLL ----+
                         |              |
                         +- NOSCROLL --+
```

Description:

If this command is issued with no parameters or with the window name parameter only, then a dialog box is opened in which the user can define the characteristics of the command line of the specified window.



Prompt> Command>

Position> Top Top or Bottom

Visible> Yes Yes or No

Lines> 1 1-9

Scroll> Yes Yes or No

Figure 130. Command Line Dialog Window.

If parameters other than the window name are supplied then the specified change is made to the command line.

Parameters:

`windowname`
: The window name containing the command line. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

`SHOW`
`HIDE`
: Show or hide the command line.

`TOP`
`BOT`
: Show the command line at the top or the bottom of the window.

`PROMPT`
: The character string following the PROMPT= keyword is used as the prompt prefixing the command line input area in the window.

`SCROLL`
`NOSCROLL`
: For CBLe only, show or hide the ISPF Edit style scroll entry field.
For INTERFACE=ISPF, the default is SCROLL.
For INTERFACE=CBLE, the default is NOSCROLL.

COMPFILE

Syntax:

```
>>- COMPFile -----+------------------------------------------------------+---><
                  |                                                      |
                  +- | Unformatted Compare Opts | -+-- | Common Opts | --+
                  |                                |
                  +- | Formatted Compare Opts | ---+
```

Unformatted Compare Opts:

```
 >-- new_fileid ------------------------ old_fileid ------------------------->

   (1)
 >--+-----------+--------------------+------------+-------------------------->
    |           |                    |            |
    |           +- STARTCol rec_pos -+            |
    |                                             |
    +---------------------+-+---------------------+
    |                     | |                     |
    +- NSTARTCol rec_pos -+ +- OSTARTCol rec_pos -+

 >--+-------------+----------------------+------------+--------------------->
    |             |                      |            |
    |             +- COMPARELEN n_bytes -+            |
    |                                                 |
    +-----------------------+-+-----------------------+
    |                       | |                       |
    +- NCOMPARELEN n_bytes -+ +- OCOMPARELEN n_bytes -+

 >--+-----------+--------------------+------------+-------------------------->
    |           |                    |            |
    |           +- STRIP -+--------+-+            |
    |                     |        |              |
    |                     +- char -+              |
    |                                             |
    +---------------------+-+---------------------+
    |                     | |                     |
    +- NSTRIP -+--------+-+ +- OSTRIP -+--------+-+
               |        |              |        |
               +- char -+              +- char -+
```

Formatted Compare Opts:

```
                             +- SDO ---+
                             |         |
 >-- new_fileid ----- USING -+---------+--- new_structname ------------------>
                             |         |
                             +- COBOL -+
                             |         |
                             +- PL1 ---+
                             |         |
                             +- ADAta -+

 >-- old_fileid --+------------------------------------------+--------------->
                  |                                          |
                  |          +- SDO ---+                     |
                  |          |         |                     |
                  +-- USING -+---------+--- old_structname --+
                             |         |
                             +- COBOL -+
                             |         |
                             +- PL1 ---+
                             |         |
                             +- ADAta -+

   (1)
 >-+----------------------+--+---------------------------------------------+->
   |                      |  |                                             |
   |        +----- , ---+ |  | +-----------------------------------------+ |
   |        V           | |  | |                                         | |
   +- VIEW -+- rectype -+-+  | |          +------ , ----+                | |
                             | v          V             |                | |
                             +-+- SELect -+- fieldname -+- FROM rectype -+-+
```

Common Opts:

```
>--+---------------+-----------------------------+----------------+--------->
   |               |                             |                |
   |               +-+- FROM -----+- start_rec --+                |
   |               | +- STARTREC -+              |                |
   |               |                             |                |
   |               +--- STARTKEY --- start_key --+                |
   |               |                             |                |
   |               +--- STARTRBA --- start_rba --+                |
   |                                                              |
   +-----------------------------+--+-----------------------------+
   |                             |  |                             |
   +-+- NFROM -----+- start_rec -+  +-+- OFROM -----+- start_rec -+
   | +- NSTARTREC -+             |  | +- OSTARTREC -+             |
   |                             |  |                             |
   +--- NSTARTKEY --- start_key -+  +--- OSTARTKEY --- start_key -+
   |                             |  |                             |
   +--- NSTARTRBA --- start_rba -+  +--- OSTARTRBA --- start_rba -+

>--+---------+--------------+--------+-----+--------------------------+----->
   |         |              |        |     |                          |
   |         +- FOR n_recs -+        |     +-| Synchronisation Opts |-+
   |                                 |
   +---------------+-+---------------+
   |               | |               |
   +- NFOR n_recs -+ +- OFOR n_recs -+

>--+-------------+-+-------------+-+--------------+-+------------------+---->
   |             | |             | |              | |                  |
   +- EXChanged -+ +- EXDeleted -+ +- EXInserted -+ +- EXFieldchanged -+

>--+--------------+---+-----------+---------------+---------------------+--->
   |              |   |           |               |                     |
   +- INCMatched -+   +- INCKeys -+               +- REPort -+- fileid -+
                                                  +- RPT ----+

>--+-----------------+--+----------------------+--+---------------------+--->
   |                 |  |                      |  |                     |
   +- LIMIT n_diffs -+  +-| Output File Opts |-+  +-- CASEIgnore -------+
                                                  +-- CASEInsensitive --+
```

Synchronisation Opts:

```
   +- SYNC -+                                +- | ReadAhead Opts | -+
   |        |                                |                      |
>--+--------+-+------------------------------+----------------------+---+--->
              |                                                         |
              +-- 1TO1 -------------------------------------------------+
              |                                                         |
              +-------------- | KEY Opts | -----------------------------+
              |                                                         |
              |                              +- | ReadAhead Opts | -+   |
              |                              |                      |   |
              +-- UNSORTED -- | KEY Opts | --+----------------------+---+
```

KEY Opts:

```
>--+--------------------------------------------------+--------------------->
   |                                                  |
   |  +--------------------------------------------+  |
   |  |                                            |  |
   |  |        +------ , ----+  +- FROM rectype -+ |  |
   |  v  (2)   V             |  |                |  |  |
   +--+-- KEY -+- fieldname -+--+----------------+-+--+
   |                                                  |
   |  +--------------------------------------------+  |
   |  |                                            |  |
   |  V                                            |  |
   +--+-- KEY --- (length  new_pos  old_pos) ------+--+
```

ReadAhead Opts:

```
               +-- ( 100          1           ) -+
               |                                 |
>-- READAhead -+---------------------------------+--+---------------+------->
               |                                 |  |               |
               +-- ( ra_limit -+-----------+- ) -+  +- SYNCONBLANK -+
                               |           |
                               +- ra_match +
```

Output File Opts:

```
>--+----------------------------------+-+---------------------------------+->
   |                                  | |                                 |
   +- WRITECHANGEDNew --+- cn_fileid -+ +- WRITECHANGEDOld -+- co_fileid -+
   +- WRITECN ----------+               +- WRITECO ---------+

>--+----------------------------------+-+---------------------------------+->
   |                                  | |                                 |
   +- WRITEMATCHEDNew --+- mn_fileid -+ +- WRITEMATCHEDOld -+- mo_fileid -+
   +- WRITEMN ----------+               +- WRITEMO ---------+

>--+----------------------------------+-+---------------------------------+->
   |                                  | |                                 |
   +- WRITEInsertednew -+- in_fileid -+ +- WRITEDeletedold -+- do_fileid -+
   +- WRITEIN ----------+               +- WRITEDO ---------+
```

Notes:

1. All parameter options that follow may be entered in any order.
2. Formatted Compare only.

Description:

Use the COMPFILE command to perform an Unformatted or Formatted compare of two files and optionally report record matches, inserts, deletions and/or changes. Execute COMPFILE with no parameters to open the Compare Files panel.

Files selected for compare may be any combination of cataloged or uncataloged sequential datasets or PDS/PDSE library members, VSAM datasets or HFS files.

File compare has been categorised as Basic (Unformatted), Extended Unformatted and Extended Formatted, each of which is discussed in detail under *File Compare* in chapter *SELCOPY/i Utilities*.

So long as the remaining command syntax is valid, any area or areas of the COMPFILE command string may be commented, and so ignored by the SELCOPY/i command parser, using REXX style comment delimiters. i.e. enclose areas of the command stream text between "/*" and "*/". This is particularly useful when COMPFILE is executed (with <PF4>) from the user's HOME (CMX) data set command centre where, using the continuation character, commands may span several lines. e.g. To temporarily omit options EXChanged and INCMatched in the following COMPFILE command...

```
<COMPF                                       \
  CBLMCT:SELCOPY.CBLI.CBLE(BOXTOT)           \
  CBL.CBLI310.CBLE(BOXTOT)                   \
/*                                           \
   EXCHANGED                                 \
*/                                           \
   STARTREC    2                             \
   STARTCOL   11                             \
   COMPARELEN 70                             \
/*                                           \
   INCMATCHED                                \
*/                                           \
   SYNC READAHEAD (100 1)
```

COMPFILE generates an output report in an SDE window view. See Compare Files Output.

Parameters:

`new_fileid`

Specifies the PDS/PDSE member, HFS file path, Sequential or VSAM data set containing the NEW image of the file data to be compared.

If *new_fileid* references an uncataloged sequential data set or PDS library, then a volume serial number prefix must be included on *new_fileid* in the format "*volser*:*data.set.name*".

new_fileid may optionally be enclosed in single quotes/apostrophes (') which get stripped by COMPFILE. Note that, regardless of whether enclosing quotes are specified the TSO prefix is never added to the *new_fileid* DSN.

`old_fileid`

Specifies the PDS/PDSE member, HFS file path, Sequential or VSAM data set containing the OLD image of the file data to be compared.

Rules for specification of *old_fileid* are the same as for *new_fileid*.

`USING SDO|COBOL|PL1|ADATA new_structname`

Specifies the structure format (SDE structure, COBOL or PL1 Copybook, COBOL or PL1 ADATA output) and structure name (PDS/PDSE library member or Sequential data set) to be used to map record data in *new_fileid*.
If no structure format is specified, the default is SDO (SDE structure).

This option forces a formatted compare of the record data.

`USING SDO|COBOL|PL1|ADATA old_structname`

Specifies the structure format and structure name used to map record data in *old_fileid* for a formatted compare. See USING *new_structname*.

Records that are assigned a record type defined in *old_structname* will be compared with records assigned the same record type in *new_structname*. Any records in the NEW and OLD files that are assigned a record type that does not exist in both *old_structname* and *new_structname* will be flagged as either inserted or deleted as appropriate.

If a field column in a record type defined in *new_structname* does not exist in the record type of the same name defined in *old_structname* or vice versa, then that field is excluded from the compare. Thus, only field columns of the same name in record types of the same name are compared.

If not specified, *new_structname* will be used to map record data in *old_fileid*.

`STARTCOL | NSTARTCOL | OSTARTCOL rec_pos`

For **unformatted compare** only, STARTCOL specifies the 1st position within the records of **both** the old and NEW files at which the data compare will begin. Record data at positions before *rec_pos* is not included in the compare.

If *rec_pos* is a position beyond the length of the record, then the compare length is always set to zero, regardless of any COMPARELEN value.

Alternatively, the record data start positions may be specified differently for the NEW and OLD files using NSTARTCOL and OSTARTCOL respectively. e.g. To compare records in NEW file starting at position 101 against records in OLD file starting at position 51.

If STARTCOL/NSTARTCOL/OSTARTCOL is not specified, *rec_pos* defaults to 1.

`COMPARELEN | NCOMPARELEN | OCOMPARELEN n_bytes`

For **unformatted compare** only, COMPARELEN specifies the maximum number of bytes *n_bytes* of data to be compared within the records of **both** the old and NEW files. The start of the data to be compared is *rec_pos* as specified by STARTCOL/NSTARTCOL/OSTARTCOL. Record data at position *rec_pos + n_bytes* and beyond is not included in the compare.

If *n_bytes* is greater than the amount of data remaining in the record, then the length of data compared is equal to 1 plus the record length minus *rec_pos*.

Alternatively, the record data maximum compare lengths may be specified differently for the NEW and OLD files using NCOMPARELEN and OCOMPARELEN respectively. Because COMPFILE will always flag a compare mismatch if the NEW and OLD compare lengths are different, specification of NCOMPARELEN and/or OCOMPARELEN is valid only when used with the STRIP/NSTRIP/OSTRIP option. e.g. To compare a blank padded character field of length 10 in OLD file against a blank padded character field of length 20 in NEW file, use `NCOMPARELEN 20 OCOMPARELEN 10 STRIP`.

If COMPARELEN/NCOMPARELEN/OCOMPARELEN is not specified, *n_bytes* defaults to 1 plus the length of the record minus *rec_pos*.

`STRIP | NSTRIP | OSTRIP char`

For **unformatted compare** only, STRIP indicates that trailing characters in the compare data of **both** the NEW and OLD files are to be stripped (so reducing the compare length) and optionally specifies the strip character *char*. For each occurrence of *char* occupying the last position of the compare data the compare length is reduced by 1.

Alternatively, a different strip *char* may be specified for compare data in the NEW and OLD files using NSTRIP and OSTRIP respectively.

If STRIP/NSTRIP/OSTRIP is not specified, then no character are stripped from the compare data. If any of these parameters are specified without *char*, the default is blank (X'40').

`VIEW rectype, ...`

For **formatted compare** only, VIEW specifies one or more record types (*rectype*) that are defined by the *new_structname* structure. Only records in the NEW and OLD files that are assigned one of these record types are selected for compare. All other records are excluded from the compare.

Note that record subsetting using parameters STARTREC/STARTKEY/STARTRBA and FOR is performed prior to the records being filtered by VIEW.

If VIEW is not specified, then all records in the NEW and OLD files are included in the compare regardless of whether they are assigned a record type in *new_structname*.

`SELECT fieldname, ... FROM rectype`

For **formatted compare** only, SELECT specifies one or more field column names (*fieldname*) to be included in the compare and the *new_structname* record types (*rectype*), to which the *fieldname* columns belong.

If a record is assigned a record type that has been selected for compare (see VIEW) and the record type matches one of *rectype* specified by SELECT, then only the *fieldname* columns will be included in the compare for that record. All other field columns are excluded.

If a record is assigned a record type that has been selected for compare and the record type does not match one of *rectype* specified by SELECT, then all field columns belonging to that record will be included in the compare.

If *fieldname* exists in *rectype* for *new_structname* but not in the *rectype* for *old_structname* then that *fieldname* is not included in the compare. If *fieldname* does not exist in *rectype* for *new_structname* then error ZZSD148E is returned.

If SELECT is not specified, then the compare includes all field columns of the same name, belonging to record types of the same name in *old_structname* and *new_structname*.

`STARTREC|FROM | NSTARTREC|NFROM | OSTARTREC|OFROM start_rec`

STARTREC (or FROM) specifies the record number *start_rec* of the first record in **both** the NEW and OLD files at which COMPFILE will begin the compare.

If *start_rec* is greater than the number of records in one file but not the other, then the remaining records in the file with the greater number of records will be flagged as inserts or deletions as appropriate. If *start_rec* is greater than the number of records in both the NEW and OLD file, then no records are compared.

Alternatively, *start_rec* may be specified differently for the NEW and OLD file using NSTARTREC (or NFROM) and OSTARTREC (or OFROM) respectively.

If STARTREC/NSTARTREC/OSTARTREC is not specified, *start_rec* defaults to 1.

`STARTKEY | NSTARTKEY | OSTARTKEY start_key`

For VSAM KSDS, VRDS files or PATHs only, STARTKEY specifies a full or partial key (*start_key*) used to identify the first record in **both** the NEW and OLD files at which COMPFILE will begin the compare.

start_key may be specified as a character or hex string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing of *start_key* will occur if specified as a character string without the "C" (or "c") prefix.

The record selected by *start_key* will be the first record with key field data which is greater than or equal to *start_key*. If STARTKEY selects a record from one file but not the other, then the remaining records in the file with a selected key record will be flagged as inserts or deletions as appropriate. If no record is selected from either file, then no records are compared.

Alternatively, *start_key* may be specified differently for the NEW and OLD file using NSTARTKEY and OSTARTKEY respectively.

If STARTKEY/NSTARTKEY/OSTARTKEY is not specified, *start_key* defaults to a number of hex zeroes (X'00') for the length of the key.

`STARTRBA | NSTARTRBA | OSTARTRBA start_rba`

For VSAM ESDS and RRDS files only, STARTRBA specifies a relative byte address (*start_rba*) used to identify the first record in **both** the NEW and OLD files at which COMPFILE will begin the compare.

start_rba may be specified as a decimal integer or hexadecimal value.

The record selected by *start_rba* will be the first record with a relative byte address which is greater than or equal to *start_rba*. If STARTRBA selects a record from one file but not the other, then the remaining records in the file containing a selected record will be flagged as inserts or deletions as appropriate. If no record is selected from either file, then no records are compared.

Alternatively, *start_rba* may be specified differently for the NEW and OLD file using NSTARTRBA and OSTARTRBA respectively.

If STARTRBA/NSTARTRBA/OSTARTRBA is not specified, *start_rba* defaults to 0.

`FOR | NFOR | OFOR n_recs`

FOR specifies the number of records (*n_recs*) in **both** the NEW and OLD files to be compared starting at the STARTREC/STARTKEY/STARTRBA record.

Record subsetting using parameters STARTREC/STARTKEY/STARTRBA and FOR is performed before any other record filtering. (e.g. using VIEW)

Alternatively, *n_recs* may be specified differently for the NEW and OLD file using NFOR and OFOR respectively.

If FOR/NFOR/OFOR is not specified, *n_recs* defaults to all remaining records in the file.

Synchronisation Opts

Following a mismatch in record data belonging to the NEW and OLD files, COMPFILE uses synchronisation options to determine the next record in the NEW file and the next record in the OLD file at which the compare will continue.

If no synchronisation method is specified, then Read-Ahead synchronisation is used. i.e. SYNC READHEAD(100 1)

Records in the NEW file that are skipped in order to perform this record synchronisation are flagged as having been **inserted**. Similarly, records that have been skipped in the OLD file are flagged as having been **deleted**.

`SYNC`

An optional keyword specified before the nominated synchronisation method.

`READAHEAD { (`*`ra_limit`*` {`*`ra_match`*`} ) }`

Applicable to both Read-Ahead and Unsorted Key synchronisation, READAHEAD specifies *ra_limit* and optionally *ra_match* values to identify the criteria used by COMPFILE to find a re-synchronisation point in the NEW and OLD files.

ra_limit is the maximum number of records that may be read in order to find a record that matches the mismatched record in the other file. Starting at the record immediately following the mismatched record pair, records are first read from the OLD file and compared with the mismatching record in the NEW file. The process is then repeated with records read from the NEW file compared with the mismathing record in the OLD file.

If a read-ahead match is found, the *ra_match* value is then used to indentify a minimum number of consecutive pairs of matching records that must exist for this to be considered a valid synchronisation point.

If a matching pair of blank records is encountered then, by default, the *ra_match* value is incremented by 1 so omitting these records from the re-synchronisation process. See option SYNCONBLANK to include matching pairs of blank records in the *ra_match* count.

For more details on Read-Ahead synchronisation, see topic File Compare.

Default for *ra_limit* is 100 and default for *ra_match* is 1.
Read-Ahead synchronosation (without key) is default for COMPFILE.

`SYNCONBLANK`

For Read-Ahead synchronisation only, COMPFILE will not include a matching pair of blank records in the *ra_match* count of consecutive matching record pairs. Therefore, a pair of matching blank lines will require a further match in the next pair of consecutive records in order to qualify the intitial match as a synchronisation point.

SYNCONBLANK will bypass this feature so that pairs of matching blank records are included in the *ra_match* count.

`1TO1`

1TO1 specifies that no synchronisation is to be performed by COMPFILE. Each mismatching pair of records is flagged as being changed.

`UNSORTED`

UNSORTED indicates that synchronisation is performed on key data within records that are not necessarily sorted in ascending key order. (See parameter KEY.)

Read-Ahead processing, as described by parameter READAHEAD, is used to synchronise records based only on data contained in the specified key fields.

`KEY`

KEY indicates that keyed synchronisation is to be performed and also defines one or more segments within the record data from which the key is comprised. Key segments are specified either as fields in a formatted record or as absolute positions and lengths within a formatted or unformatted record.

See File Compare for details of sorted and unsorted key synchronisation.

`KEY `*`fieldname, ...`*` {FROM `*`rectype`*`}`

For formatted compare only, specifies one or more field names *fieldname* (or field references) that belong to the nominated record type *rectype*. Each of these fields constitute a key segment within records that are assigned *rectype*.

If a different structure is applied to OLD file data (i.e. USING *old_structname*), then each key *fieldname* must exist in record type *rectype* for both *new_structname* and *old_structname*.

The KEY parameter may be specified repeatedly to define key field segments in more than one record type. If FROM *rectype* is not specified, then the first record type defined in the structure (SDO) is assumed.

`KEY (`*`length new_pos old_pos`*`)`

For formatted or unformatted compare, this form of the KEY parameter expression identifies a key segment length (*length*) and positions within the new (*new_pos*) and old (*old_pos*) file records.

The KEY parameter may be specified repeatedly to define multiple corresponding key segments in the NEW and OLD file records. These key segments apply to all records in the NEW and OLD files. i.e. For formatted record compare, the same key is applied to records assigned any record type in their unformatted state.

`EXCHANGED`

EXCHANGED indicates that report record types Compare or Compare *record_type* which specifically describe changed records from the NEW (CN) and OLD (CO) files, are to be excluded from the output report.

Default is to include these records in the report.

`EXDELETED`

EXDELETED indicates that report record types Compare or Compare *record_type* which specifically describe records that have been deleted (D) from the OLD file, are to be excluded from the output report.

Default is to include these records in the report.

`EXINSERTED`

EXINSERTED indicates that report record types Compare or Compare *record_type* which specifically describe records that have been inserted (I) into the NEW file, are to be excluded from the output report.

Default is to include these records in the report.

`EXFIELDCHANGED`

Applicable to formatted compare only, EXFIELDCHANGED indicates that report record types Compare or Compare *record_type* which specifically identify the names of fields containing changed data (C), are to be excluded from the output report.

By default, for formatted compare only, a record flagged as changed will produce one or more of changed field report records each identifying the name of a field containing changed data. If EXCHANGED is not specified, these report records immediately follow the CN/CO report records.

Note that, specifying EXFIELDCHANGED may result in a significant performance improvement since the process of comparing field-by-field is terminated at the first mismatch in each record.

`INCMATCHED`

INCMATCHED indicates that report record types Compare or Compare *record_type* which specifically describe matching record pairs, are to be included in the output report.

Default is to exclude these records from the report.

`INCKEYS`

INCKEYS indicates that, for sorted and unsorted KEY synchronisation only, records containing a matching key field will be displayed in the output report, regardless of whether INCMATCHED has been specified.

This is of primary use when performing a formatted compare of hierarchical records where records are assigned to different record types and keys fields are defined in multiple record types.

Default is to exclude these records from the report.

`LIMIT` *`n_diffs`*

LIMIT specifies a number of mismatches *n_diffs* that may occur before COMPFILE is terminated. If *n_diffs* is 0 (zero) or LIMIT is not specified, then there is no limit to the number of differences and COMPFILE only terminates after all records selected for compare have been processed.

`REPORT|RPT` *`fileid`*

REPORT specifies that the COMPFILE report is to be written to the specified sequential data set or PDS/PDSE member *fileid*.

If *fileid* exists but is uncataloged, then include the required volser as part of the fileid specification in the format *volser*:*data.set.name*.

The report is a structured data file designed to be browsed (not printed) using a SELCOPY/i structure definition object (SDO), which is also generated by COMPFILE. The associated SDO fileid is constructed simply by adding '.SDO' to the DSN of the sequential or partition dataset name specified by *fileid*. The DSN is therefore restricted to 40 bytes in length. e.g. If *fileid* is ZX1234.SELCOPYI.COMPF.REPORT(XYZ001), the allocated SDO is ZX1234.SELCOPYI.COMPF.REPORT.SDO(XYZ001).

If the report file *fileid* and/or the SDO file do not already exist, then they will automatically be allocated by COMPFILE relying on SMS ACS to select a suitable storage group of eligable DASD volumes. File *fileid* is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If REPORT is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

Output File Opts

Records from the NEW and OLD files may be copied to different output files based on their status flag as assigned by COMPFILE.

The fileid specified in each of the following options must belong to a sequential data set, PDS/PDSE member or HFS file which is not already assigned an exclusive ENQ. If the output file is non-HFS and is not yet allocated, then COMPFILE will allocate the file as a cataloged data set with space allocation of TRACKS(150,75) and data set geometry modelled on the NEW or OLD file as appropriate.

If the specified output file exists but is uncataloged, then include the required volser as part of the fileid specification in the format *volser*:*data.set.name*.

COMPFILE will fail with error ZZSD027E if the same fileid is specified on more than one of the following output file options.

`WRITECHANGEDNEW|WRITECN` *`cn_fileid`*

Indicates that records from the NEW file, flagged as having been changed (CN), are to be copied to *cn_fileid*.

`WRITECHANGEDOLD|WRITECO` *`co_fileid`*

Indicates that records from the OLD file, flagged as having been changed (CO), are to be copied to *co_fileid*.

`WRITEMATCHEDNEW|WRITEMN` *mn_fileid*

Indicates that records from the NEW file, flagged as matching records in the OLD file, are to be copied to *mn_fileid*.

`WRITEMATCHEDOLD|WRITEMO` *mo_fileid*

Indicates that records from the OLD file, flagged as matching records in the NEW file, are to be copied to *mo_fileid*.

`WRITEINSERTEDNEW|WRITEIN` *in_fileid*

Indicates that records from the NEW file, flagged as having been inserted (I), are to be copied to *in_fileid*.

`WRITEDELETEDOLD|WRITEDO` *do_fileid*

Indicates that records from the OLD file, flagged as having been deleted (D), are to be copied to *do_fileid*.

`CASEIGNORE|CASEINSENSITIVE`

Both CASEIGNORE and CASEINSENSITIVE specify that the compare of record data treats a lower case alpha character as being a match with its equivalent upper case character. For formatted compare, this option applies to character (AN) fields only.

Parameters CASEIGNORE and CASEINSENSITIVE do not affect key synchronisation where alpha characters in the key segments must match exactly.

Examples:

Use of the COMPFILE command may result in long command streams. Therefore, it is recommended that any COMPFILE command you enter (or generate via the Compare File dialog panel) should be entered as text in your HOME command centre (CMX) data set. Doing this will save retyping the command in order to make any required syntax changes before re-execution.

```
<compf  CBL.CBLI310.ASM(WINFIPO0)  CBL.CBLI190.ASM(WINFIPO0)  sync readahead(100 1)
```

Compare two versions of the same Assembler source module to identify record changes, inserts and deletions that have been made between program releases. Read-Ahead synchronisation is used since code insertions/deletions are expected.

```
<compfile    CBL.SELC300.SZZSMAC(ZZSNAME)                       \
             CBL.SELC300.AZZSMAC(ZZSNAME)                       \
             sync readahead(40 1)                               \
             nstartrec 10      ostartrec 15      for 100        \
             startcol 1   comparelen 72
```

Compare the target and distribution library copies of the SELCOPY Product Suite CBLNAME macro to identify whether differences exist. Macro contains additional 5 lines of comment data in the distribution (OLD file) library so different STARTREC values are required (OSTARTREC/NSTARTREC). Compare excludes sequence numbers that may exist in columns 73-80 using a STARTCOL/COMPARELEN combination.

```
<compfile    NBJ2849.AMCUST.G01647.DA                           \
             NBJ2849.AMCUST.G01646.DA                           \
             using  cobol  MAST649.AM.COB(AMCUST)               \
             view    ADDR_REC                                   \
             select  CUSTID, ADDR1, POSTCODE  FROM ADDR_REC     \
             key     CUSTID                   FROM ADDR_REC
```

Compare NEW and OLD versions of a customer records file containing records mapped by a cobol copybook record structure (01 level), ADDR_REC. The records are sorted in ascending order on key field, CUSTID, which is used here for re-synchronisation of corresponding record pairs. Only fields CUSTID, ADDR1 and POSTCODE are compared.

COMPLIB

Syntax:

```
>>-- COMPLIB ---------+-------------------+-----------------------------------><
                      |                   |
                      +- | CLI Options | -+
```

CLI Options:

```
 >-- new_libdsn --+------------------------+-------- old_libdsn -------------->
                  |                        |
                  |     +------------+     |
                  |     v            |     |
                  +- ( --- mbr_mask -+- ) -+

 >-------+--------------------+----+---------+-------------------------------->
         |                    |    |         |
         +- STRIP -+----------+    +- BATCH -+
                   |          |
                   +- 'char' -+
```

Description:

The COMPLIB command invokes the Compare Libraries utility to perform a compare of two PDS/PDSE libraries in the foreground, generate a batch job to perform the compare or, if executed with no parameters, open the Compare Libraries Panel.

COMPLIB executes the SELCOPY program to compare selected library members and generates an output report suitable for edit using the SELCOPY/i (CBLe) text editor.

Parameters:

new_libdsn
: Specifies the full DSN of the PDS/PDSE library containing the NEW images of library member data to be compared.

mbr_mask
: Specifies a member name mask which is to be used by COMPLIB to select members from the **NEW library only** (*new_libdsn*) to be included in the compare operation. Members of the OLD library (*old_libdsn*), whose member names match a specified *mbr_mask*, are **not** selected for compare unless a member of the same name exists in the NEW library.

mbr_mask may optionally contain the following wild card characters:

- `*` A single asterisk represents an entire member name or zero or more characters within a member name mask.
- `%` A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS/PDSE *new_libdsn* and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by one or more blanks and/or a "," (comma).

Default is to compare **all** members of both the NEW and OLD libraries.

old_libdsn
: Specifies the full DSN of the PDS/PDSE library containing the OLD images of library member data to be compared.

STRIP < *char* >
: STRIP indicates that trailing characters in the compare data are to be stripped from the longer record, to the length of the shorter record, when the records to be compared are of different lengths. The optional strip character *char* may be specified in quotes (") or apostrophes (').

If STRIP is not specified, then no characters are stripped from the compare data. If STRIP is specified without *char*, the default strip character is blank ' ' (X'40').

BATCH
: BATCH specifies that COMPLIB should generate JCL to execute SELCOPY in batch. Concatenated SYSIN input includes the STRIP option status and strip character, the input control statements (ZZSCOMPL) and, if *mbr_mask* was specified, a list of selected members. By default, SYSPRINT is written to SYSOUT=*.

The JCL will be displayed in a temporary CBLe text edit window view.

If BATCH is not specified, COMPLIB will execute SELCOPY in the foreground to generate the compare libraries utility output.

Examples:

```
COMPLIB  NBJ.CBLI310.ASM  NBJ.CBLI31B.ASM
```

Compare all members of PDS library NBJ.CBLI310.ASM with all members of PDS library NBJ.CBLI31B.ASM.

```
COMPLIB  NBJ.SMPEINST.JCL(RX* QS*)  NBJ.SMPEINST.JCL.BKUP  BATCH
```

Generate a batch SELCOPY job to compare all members with names beginning "RX" or "QS" in PDS library NBJ.SMPEINST.JCL with equivalent members of PDS library NBJ.SMPEINST.JCL.BKUP.

CURSORSELECT

Syntax:

```
>>--+-- CURSORSELECT --+----------------------------------------------------><
    |                  |
    +-- CSELECT -------+
    |                  |
    +-- CSEL ----------+
```

Description:

CURSORSELECT is a generic command that performs the default operation for the field at the cursor position.

The default operation depends on the function of the window, the window class and the field in which the cursor is positioned. The default operation for a particular window/field may be found in the documentation for the relevant window type.

CURSORSELECT is intended to be assigned to a function key to execute the same operation as that performed when the >Enter< key is hit.

DB2

Syntax:

```
            +--------------------------------------------------+
            v                                                  |
>>-- DB2 ------+-------------------------+---+--------------+--+------------><
               |                         |   |              |
               |  +-------------------+  |   +-- fastpath --+
               |  v                   |  |
               +---- fieldname=value -+--+
```

Description:

Display the SELCOPY/i DB2 primary option menu panel and, optionally, nested sub-panels. If a fast path is specified or, if no fast path is specified but a value is specified for field SSN (SSN=*xxxx*), then a DB2 connection is performed automatically. In all other cases, no DB2 connection is attempted until the user hits <Enter> (or any PFKey) to accept the DB2 SSN and SQLID displayed in the DB2 primary option menu panel.

Parameters:

`fastpath`

Each SELCOPY/i DB2 panel may be accessed via one or more levels of nesting into menu panels. Fast path menu item selection allows immediate nesting into panel menus to open the required SELCOPY/i DB2 panel without the need to select menu items individually.

Note that each nested panel in the fast path will be displayed. Furthermore, a connection to the specified subsystem (SSN=*xxxx*) or the last accessed or default DB2 subsystem, will occur automatically with SQLID set to the corresponding value in the DB2 primary option menu panel.

fastpath is an integer value representing a menu choice. If no update of input fields (*fieldname*=*value*) is to be performed between successive *fastpath* values, then successive *fastpath* values may be separated by a single dot/period (.) without intervening blanks.   e.g. DB2 7.1

`fieldname=value`

fieldname nominates a field name within the current nested DB2 panel, into which the corresponding *value* will be inserted. If *value* contains special characters or blanks, then it must be contained within apostrophes (') or quotation marks

(").

A separate *fieldname=value* combination may be specified for each input field in the resulting panel.

Beware that specifying a fast path in addition to a *fieldname=value* combination will immediately execute the command assigned to the resultant panel.   e.g. CREATE STOGROUP, DROP TABLE.

Examples:

```
DB2  5
```

Open the CREATE Object option panel.

```
DB2  SQLID=NBJ002
```

Open the DB2 primary option menu panel and insert "NBJ002" in the SQLID field. Connection will **not** occur when the panel is opened until <Enter> or a PFKey is hit.

```
DB2  SSN=DB2B  1  DB2CMD="-DISPLAY BUFFERPOOL(BP1)"
```

Open the DB2 primary option menu panel, insert "DB2B" in the SSN field and perform DB2 connect. Then open the DB2 Command panel, insert "-DISPLAY BUFFERPOOL(BP1)" in the DB2CMD field, and immediately execute the command to display a list of DB2 buffer pools.

```
DB2  SSN=DB2C   7.1   NAME=DSN%
```

Open the DB2 primary option menu panel, insert "DB2C" in the SSN field and perform DB2 connect. Then open the List Storage Groups panel, insert "DSN%" in the NAME field, and immediately execute to display a list of storage groups with name beginning "DSN".

DCMD

Syntax:

```
>>-- DCMD -+-------------------+--+------------------+--+-----------------+-->< 
           |                   |  |                  |  |                 |
           +-- -SSN ssn_name ---+  +-- -LIMIT n_bytes -+  +-- db2_command ---+
```

Description:

Use the DCMD command to open the Execute DB2 Commands (ZZS2XDB2) panel.

If the user has already opened a DB2 Primary Option Menu in the current SELCOPY/i session and so a DB2 connection already exists for the specified DB2 subsystem, then no new connection is made. Otherwise, a connection is made to this DB2 subsystem when the panel is opened.

Note: Not implemented for CMS or VSE.

Parameters:

`-SSN ssn_name`

The DB2 subsystem in which the specified DB2 command will be executed.

Default is the last DB2 subsystem to which the user connected in this or previous SELCOPY/i sessions. If no previous DB2 connection has been made using SELCOPY/i, the default is the value of the SELCOPY/i INI option, DB2.SSN, or else the subsystem name specified in the DB2SubSys field of the CBLNAME load module.

`-LIMIT n_bytes`

Limits the number of bytes used for the DB2 command output data buffer.

Where the length of data returned by the command exceeds the output buffer size, then error message ZZSX016W is returned indicating the number of bytes of output data returned, and number of bytes not returned by the command.

The *n_bytes* value is placed in the "**Byte Limit>**" field of the Execute DB2 Command panel and the default limit is 0 (no limit).

`db2_command`

A valid DB2 command to be executed when the Execute DB2 Command panel is opened.

The *db2_command* string is placed in the "Command>" field of the Execute DB2 Command panel and the default is a null string.

Examples:

```
DCMD -SSN DB8G  -DISPLAY BUFFERPOOL(BP0)
```

Display the status of buffer pool BP0 in subsystem DB8G.

DOWN

Syntax:

```
>>-- DOWN --+--------------+--+-----------+---------------------------------><
            |              |  |           |
            +- windowname -+  +- CURSOR --+
                              |           |
                              +- DATA ----+
                              |           |
                              +- HALF ----+
                              |           |
                              +- MAX -----+
                              |           |
                              +- PAGE ----+
                              |           |
                              +- n_lines -+
```

Description:

Scroll the view of the data within the specified window downwards towards the bottom of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_lines* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, DOWN, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to DOWN is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one line is used.

By default this command is assigned to function key **PF8**.

Parameters:

`windowname`
The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

`CURSOR`
The line on which the cursor is positioned becomes the first line of the scrolled display.
If the cursor is positioned outside the display area or on the first line within the display area, then DOWN PAGE is executed instead.

`DATA`
Scroll down to display one page (display window depth) less one line of data.
The last line in the current display area becomes the first line of the scrolled display.

`HALF`
Scroll down half a page of data.
The line that is half way down the page of data in the current display area becomes the first record of the scrolled display.

`MAX`
Scroll down to display the last page of data.
Where more than one page of data exists, the last displayable line becomes the last line of the scrolled display.
Otherwise, the first line of data becomes the first line of the scrolled display.

`PAGE`
Scroll down to display the next whole page of data.
The line following the last line of the current display area becomes the first line of the scrolled display.

`n_lines`
Scroll down a specified number of lines.
The line that is *n_lines* below the current line becomes the first line of the scrolled display.

DSQL

Syntax:

```
>>-- DSQL -+-------------------+--+-----------------+--+-----------------+--->< 
           |                   |  |                 |  |                 |
           +- -SSN ssn_name ---+  +- -LIMIT n_rows -+  +- sql_syntax ----+
```

Description:

Use the DSQL command to open the Execute SQL Statements (ZZS2XSQL) panel.

If the user has already opened a DB2 Primary Option Menu in the current SELCOPY/i session and so a DB2 connection already exists for the specified DB2 subsystem, then no new connection is made. Otherwise, a connection is made to this DB2 subsystem when the panel is opened.

Note: Not implemented for CMS or VSE.

Parameters:

`-SSN` *`ssn_name`*
: The DB2 subsystem in which the specified SQL statement will be executed.

Default is the last DB2 subsystem to which the user connected in this or previous SELCOPY/i sessions. If no previous DB2 connection has been made using SELCOPY/i, the default is the value of the SELCOPY/i INI option, DB2.SSN, or else the subsystem name specified in the DB2SubSys field of the CBLNAME load module.

`-LIMIT` *`n_rows`*
: Limit the number of rows to be displayed in the display area of the Execute SQL Statements panel following a SELECT transaction. Once the limit threshold has been reached, no further attempt is made to retrieve selected rows of data.

The *n_rows* value is placed in the "**Row Limit>**" field of the Execute SQL panel and the default limit is 0 (no limit).

`sql_syntax`
: Valid SQL syntax to be executed when the Execute SQL panel is opened.

The *sql_syntax* string is placed in the "Statement>" field of the Execute SQL panel and the default is the last SQL statement executed from this panel. If the default *sql_syntax* string is used, its execution is delayed.

Examples:

```
DSQL -SSN DB8G  -LIMIT 200   SELECT * FROM DSN810.EMP
```
Display the first 200 entries in the table DSN810.EMP of DB2 subsystem DB8G.

```
DSQL -SSN CBLA    GRANT EXECUTE ON PLAN CBLPLAN1 TO USER002
```
Execute an SQL GRANT statement in subsystem CBLA.

DRAGBORDERMINUS

Syntax:

```
>>--- DRAGBORDERMINUS -------------------------------------------------------><
```

Description:

Window resizing command, DRAGBORDERMINUS is intended to be assigned to PFKeys at the "Border" level (see CLI command KEYS).

DRAGBORDERMINUS moves the window's border closer to the top left corner of the 3270 display area.

When the cursor is positioned on a horizontal border, the border is dragged one row upwards; when on a vertical border, the border is dragged one column to the left; and when on a corner, the border is dragged both one row upwards and one column to the left.

The border will not be positioned outside the display area in which it is defined. i.e. MDI child window borders cannot be dragged outside its parent's client area and all other windows cannot be dragged outside the 3270 display area.

Similarly, SELCOPY/i will not allow the window borders to be dragged so that the window is smaller than 3 rows x 8 columns.

By default, DRAGBORDERMINUS is assigned to Border PFKeys F7 and F10.

DRAGBORDERPLUS

Syntax:

```
>>--- DRAGBORDERPLUS -------------------------------------------------------><
```

Description:

Window resizing command, DRAGBORDERPLUS is intended to be assigned to PFKeys at the "Border" level (see CLI command KEYS).

DRAGBORDERPLUS moves the window's border away from the top left corner of the 3270 display area.

When the cursor is positioned on a horizontal border, the border is dragged one row downwards; when on a vertical border, the border is dragged one column to the right; and when on a corner, the border is dragged both one row downwards and one column to the right.

The border will not be positioned outside the display area in which it is defined. i.e. MDI child window borders cannot be dragged outside its parent's client area and all other windows cannot be dragged outside the 3270 display area.

Similarly, SELCOPY/i will not allow the window borders to be dragged so that the window is smaller than 3 rows x 8 columns.

By default, DRAGBORDERPLUS is assigned to Border PFKeys F8 and F11.

EDIT

Syntax:

```
>>-- Edit --+------------+--+----------------------------------------------+-><
            |            |  |                                              |
            +-- fileid --+  +- ( -+-- PROFILE macroname --+-+--------------+
                                  |                       | |              |
                                  +-- NOPROFile ----------+ +-| HFS Opts |-+
```

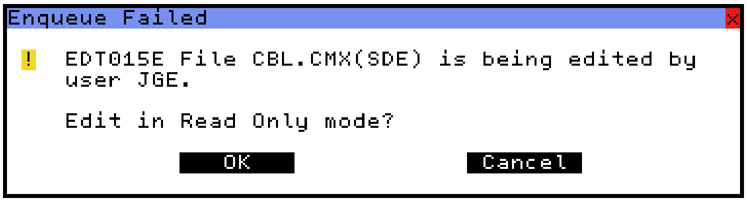
Description:

Use the EDIT command to open a CBLe text edit window view to edit a file (read/write).

If the CBLe text editor main window has been stopped, EDIT will start the CBLe main window before opening the edit view of the requested file.

If the file has a VSE LIBR LOCK applied or an exclusive MVS SPFEDIT ENQ to a different job id, then the **Enqueue Failed** pop-up window is displayed containing message ZZSE015E prompting the user to edit the file in read-only mode.

Read-only edit may be invoked directly using the VIEW command thus avoiding the Enqueue Failed window.



Enqueue Failed

! EDT015E File CBL.CMX(SDE) is being edited by user JGE.

Edit in Read Only mode?

Figure 131. Enqueue Failed Window.

Parameters:

fileid
The fileid of the file to be viewed.

For MVS, *fileid* may be the DSN of a sequential or VSAM data set, the member name (with or without the library DSN) of a PDS/PDSE member or an HFS file name. If member name is specified without a library DSN, the DSN of the library member in the current edit view is used.

For VSE, *fileid* is the member name of a LIBR library in lib.sublib.mn.mt format.

For CMS, *fileid* is a CMS fileid.

Note that if **fileid** is not specified, then the SELCOPY/i INI variable System.CmdTEXT is used. If System.CmdTEXT has not been set, then no action is taken.

`PROFILE` *`macroname`*

The REXX edit macro to be executed as the profile when editing the file.

This macro overrides use of the default profile macro defined by the SELCOPY/i INI option Edit.DefProfile=macroname and/or the CBLe command SET DEFPROFILE (default PROFILE.)

The macro name must exist in a library within the CBLe macro path.

The PROFILE option only affects the profile for the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the edit ring later in your edit session.

`NOPROFILE`

Suppresses use of a profile macro when editing the file.

The NOPROFILE option only affects the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the ring later in your edit session.

HFS Opts

See CBLe CLI command EDIT for supported HFS parameters.

Examples:

```
EDIT     NBJ.DATA.SET    (PROFILE  PROFTEMP
```

Edit NBJ.DATA.SET using macro PROFTEMP as a profile.

```
EDIT     NBJ.JCL(CBLINS01)   (NOPROF
```

Edit PDS member NBJ.JCL(CBLINS01) without a profile macro.

EO

Syntax:

```
>>--- EO --- jnm -- jno -- class --+-------------------------+--><
                                   |                         |
                                   +-- userid --+------------+
                                                |            |
                                                +-- passwd --+
```

Description:

Use the EO command to Edit (read only) an Output listing from the system's output queues. A new CBLe text editor window is opened if SELCOPY/i INI variable Edit.Instance=Multiple or if no CBLe window is already open.

A temporary fileid is used to edit the file. For VSE, the fileid is **SYSLST.class.jnm.jno**.

The SELCOPY/i command, EO, may be used in conjunction with the CBLe command, SUBMIT. A job may be submitted to batch from within CBLe and the output retrieved via EO.

In VSE, Basic Security Manager (BSM) does not impose security on the VSE POWER queues. Therefore, in order to impose access restrictions on LST queue output when SELCOPY/i INI variable System.VSESMLogon=Yes, the following restrictions apply:

1. If an entry is **not** password protected, then it may only be edited if the TO or FROM attributes match the user's userid.
2. If an entry is password protected, then it may be edited by any user so long as the password is supplied.

If SELCOPY/i INI variable System.VSESMLogon=No (i.e. no Security Manager is active), then EO is only successful if the entry is password protected and the password is specified as a parameter to EO.

Note: Not yet implemented for MVS.

Parameters:

`jnm`

The required Job Name.

`jno`

The required Job Number.
Note that, when a job is submitted using the CBLe SUBMIT command, the job number is displayed in the job submitted

confirmation message.

class
The required List Class.

userid
The userid of the user that owns the job. For VSE, this must be the userid on either the TO or FROM LST queue attributes.
Default is the current user's userid.

passwd
The password to be used when editing a password protected queue entry.
If the entry is password protected and no password is specified, then EO will fail.

Examples:

`eo  LIBRDEL  1551  S`
View list output for job LIBRDEL having job number 1551 and belonging to list class S. For successful operation, the job must have TO or FROM LST queue attributes equal to the userid of the current user and SELCOPY/i INI variable System.VSESMLogon=Yes.

`eo  CICSICCF  201  A  SYSA  SECRET`
View list output for job CICSICCF having job number 201, a TO or FROM attribute of SYSA and belonging to list class A. The queue entry is password protected with password "SECRET".

ERASE

Syntax:

Erase an MVS data set, HFS file or PDS(E) member:

```
>>-- ERAse -----+---------------------+---- fileid --------------------------><
                |                     |
                +- volid ------ : ----+
```

Erase a CMS file on an accessed minidisk:

```
>>-- ERAse -------------------------------- fileid --------------------------><
```

Erase a VSE sequential or VSAM file:

```
>>-- ERAse -----+- volid ---+-- : --------- fileid --------------------------><
                |           |
                +- catdsn --+
```

Description:

Erase (delete) a single sequential DASD file, HFS file, PDS(E) member or VSAM file.

To succeed, the user must have sufficient read/write authority for the file and no exclusive ENQ or LOCK should already exist for the file.

For VSE, sequential files may only be erased if the CBL software product **CBLVCAT** is licensed. SELCOPY/i uses CBLVCAT's DEL operation to perform the erase.

Parameters:

volid
For MVS uncataloged data sets and VSE sequential disk files, this is the volume serial number of the DASD volume on which the file resides.

catdsn
For VSE VSAM files, this is the full fileid of the VSAM catalog to which the VSAM managed file belongs.

fileid
The full fileid of the file to be erased.

For MVS, specification of a leading "." (dot/period) or "/" (slash) identifies **fileid** as being an absolute or relative HFS path name. Erasing an HFS path name performs a USS UNLINK operation for the individual HFS path name and so alternate path names to the same data are unaffected.

If *fileid* is a defined ALIAS for a non-VSAM data set, the ALIAS will be deleted, **not** the related data set.

Examples:

```
erase   test.exec.a
```

Erase CMS file TEST.EXEC.A.

```
erase      cbl.cbli.test.file
```

Erase MVS cataloged data set CBL.CBLI.TEST.FILE.

```
erase  cbl.cbli.testlib(example1)
```

Erase MVS PDS(E) member CBL.CBLI.TESTLIB(EXAMPLE1).

```
erase  OEM001:cbl.cbli.test.file
```

Erase MVS uncataloged data set CBL.CBLI.TEST.FILE from DASD volume OEM001.

```
erase   SYSWK1:CBL.SELCOPY.NAM
```

Erase VSE sequential file CBL.SELCOPY.NAM on SYSWK1. (CBLVCAT must be licensed.)

```
erase  VSESP.USER.CATALOG:CBL.TEST.KSDS
```

Erase VSE VSAM managed data set CBL.TEST.KSDS cataloged in the VSAM catalog, VSESP.USER.CATALOG.

FAV

Syntax:

```
>>---- FAV -----------------------------------------------------------------><
```

Description:

The FAV command may be used to open a Favourites Datasets/Commands window to easily access commonly used files and commands.

The dialog window will be opened with fields populated with parameters entered by the user during the last invocation of the window.

Parameters:

FAV has no parameters.

FCOPY

Syntax:

```
>>-- FCopy -----------+--------------------+---------------------------------><
                      |                    |
                      +--| CLI Options | --+
```

CLI Options:

```
>>-- from_fileid -------+------------------------------------+--------------->
                        |                                    |
                        |         +- SDO ---+                |
                        |         |         |                |
                        +- USING -+---------+-- in_struct ---+
                                  +- COBOL -+
                                  +- PL1 ---+
                                  +- ADAta -+

 >-- to_fileid ---------+------------------------------------+--------------->
                        |                                    |
                        |         +- SDO ---+                |
                        |         |         |                |
                        +- USING -+---------+-- out_struct --+
                                  +- COBOL -+
                                  +- PL1 ---+
                                  +- ADAta -+

 >--+-------------------------+-+----------------------+-+-----------------+->
    |                         | |                      | |                 |
    +- STARTKEY -- start_key -+ +- STOPAFT -+- n_recs -+ +- FILL -+- char -+
    |                         | |           |            |        |
    +- STARTRBA -- start_rba -+ +--FOR -----+            +- PAD --+
    |                         |
    +- STARTREC +- start_rec -+
    |           |
    +- FROM ----+

 >--+----------+--+---------+--+----------+--+---------+--+-------+----------><
    |          |  |         |  |          |  |         |  |       |
    +- APPend -+  +- JCL ---+  +- REPlace +  +- Quiet -+  +- NEW -+
                  |         |
                  +- BATch -+
```

Description:

The FCOPY command invokes the File Copy utility to copy a file or library members in the SELCOPY/i foreground, generate JCL to perform the copy in batch or, if executed with no parameters other than a source fileid, open the File Copy Panel.

FCOPY supports any combination of HFS files; cataloged sequential, GDG and VSAM data sets; cataloged PDS/PDSE libraries and library members as its source and target files,

The source and target file may be of different data set organisation and geometry (RECFM, LRECL, BLKSIZE) with truncation or padding of records being performed as required. e.g. Multiple members of a PDS/PDSE library may be copied to a sequential data set. VSAM KSDS records may be copied to a sequential data set or PDS/PDSE library member or HFS file etc.

Library copy is performed if the source file is a PDS/PDSE library, specified with or without a member mask, and the target file is a PDS/PDSE library with no member name specified. Note that a target PDS/PDSE library DSN with no member name is valid only for library copy.

Where possible, the File Copy utility invokes IEBCOPY to perform library copy. For file copy or where library copy performs copy functions not supported by IEBCOPY (e.g. libraries are of different geometry or the parameter STARTREC or FOR has been specified), then the SELCOPY/i File Search/Update/Copy/Remap Utility (FSU) is invoked.

Parameters:

`from_fileid`

The full filied of the source file or library to be copied.

For PDS/PDSE libraries, multiple blank and/or comma separated member masks may be specified in parentheses "( )" following the library DSN. Each member mask may contain wildcard characters, "*" (representing zero or more characters) and/or "%" (representing exactly one character.) If a library DSN is specified without a member mask, then a member mask of (*) is assumed. i.e. all members.

`USING SDO|COBOL|PL1|ADATA in_struct`

Specifies *in_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map record data fields in *from_fileid* for use in a SELCOPY/i FSU Formatted File Remap operation. Therefore, specification of an input structure *in_struct* is redundant if no output structure *out_struct* is specified for *to_fileid*.

All input records are treated as comprising a number of data fields of pre-determined lengths and of various data types, each field being identified via a field name.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. This is an overhead which may be overcome by generating a non-temporary SELCOPY/i SDO file and referencing that instead. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

When *from_fileid* is read, each input record is assigned a record type (RTO), as defined in the specified or generated SDO, and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

Data within the input record fields is remapped as described below under USING *out_struct* .

SDO, COBOL, PL1 or ADATA identifies the source format of the input structure file.
Default is SDO.

SDO	A SELCOPY/i Structured Data Object. This is the format required by SELCOPY/i to format structured data.
COBOL	A COBOL copy book. SELCOPY/i will use the COBOL compiler to compile the file in order to generate a temporary SDO.
PL1	A PL1 include file. SELCOPY/i will use the PL1 compiler to compile the file in order to generate a temporary SDO.
ADATA	An ADATA file created by a previous COBOL or PL1 compilation of a copybook/include file. SELCOPY/i will use the ADATA file to generate a temporary SDO.

`to_fileid`

The full fileid of the target file or library.

If *from_fileid* is a library and *to_fileid* is a not, then records from all selected members will be copied to the single target data set.

If *from_fileid* contains a library member mask and *to_fileid* is a library (i.e. library copy), then specification of a target library member name is invalid and FCOPY will fail. Note that FCOPY does not support specification of a target library member mask to perform a rename of multiple source members.

`USING SDO|COBOL|PL1|ADATA out_struct`

Specifies *out_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map record data fields in *to_fileid* for use in a SELCOPY/i FSU Formatted File Remap operation. Therefore, specification of an output structure *out_struct* is redundant if no output structure *in_struct* is specified for *from_fileid*.

During the remap process, the following will occur:

1. Input records of record type not defined in the output structure are copied without field remap.
2. Output structure record types not defined in the input structure are redundant and so are ignored.
3. Record data in input fields are copied to output fields of the same name belonging to record types of the same name.
4. The input field data will be reformatted to the data type of the output field and will be moved to the output field's position within the record map.
5. Any input fields whose field names are not part of the output record structure, will not be included in the output record.
6. Any output fields whose field names are not part of the input record structure are initialised to their default values.

See the input structure USING *in_struct* field for description of output USING field sub-parameters and implementation of a structure on record data.

`STARTKEY start_key`

If *from_fileid* is a VSAM KSDS, VRDS file or PATH, STARTKEY may be used to specify a full or partial key *start_key* used to identify the first source record to be copied. All records occurring before *start_key* are bypassed.

start_key may be specified as a character or hex string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing of *start_key* will occur if specified as a character string without the "C" (or "c") prefix.

The record selected by *start_key* will be the first record with key field data which is greater than or equal to *start_key*.

`STARTRBA start_rba`

If *from_fileid* is a VSAM ESDS, STARTRBA may be used to specify a relative byte address *start_rba* used to identify the first source record to be copied. All records occurring before *start_rba* are bypassed.

start_rba may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

The record selected by *start_rba* will be the first record with a relative byte address which is greater than or equal to *start_rba*.

`STARTREC start_rec`
`FROM`

STARTREC (or FROM) specifies the record number *start_rec* of the first source record to be copied from the *from_fileid* file or library member(s). All records occurring before *start_rec* are bypassed.

start_rec may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

If STARTREC is not specified, *start_rec* defaults to 1.

STOPAFT *n_recs*
FOR

STOPAFT (or FOR) specifies the maximum number of records *n_recs* to be copied from the *from_fileid* file or library member(s)

If STOPAFT is not specified, *n_recs* is unlimited.

FILL *char*
PAD

The character *char* to be used to pad short source records when copying to a longer, fixed length target record or the trailing pad character that will be stripped from fixed length records when copied to records of variable lengths.

char may be specified as a quoted character string (e.g. 'a') or as a hexadecimal string (e.g. x'00').
Default is blank (X'40').

APPEND

Applicable to file copy or remap only where *to_fileid* is not a library member, APPEND specifies that records written to the target file are to be appended to records that already exist in the target file.

Beware that, if the DSORG of the target file is a reuseable VSAM data set (IDCAMS DEFINE REUSE), then selecting NO will overwrite **all** existing records. An attempt to overwrite an existing record will fail if the VSAM data set is defined with NOREUSE.

Selecting YES will append records to the end of the target file unless it is a VSAM KSDS file. Copy to a VSAM KSDS file will write records to their correct key positions within the data set. If a record is not in key sequence or contains a duplicate key, then that record will not be copied and the copy operation continues at the next source file record.

If this option is not selected, then the existing records will be overwritten.

JCL
BATCH

Do not execute the copy immediately but instead generate JCL in a temporary edit view that executes program SDEAMAIN to run the FCOPY command in batch.

REPLACE

Applicable to library copy or remap only, indicates that members that exist in *to_fileid* will be replaced by input members of the same name.

If this option is not selected, then existing members will not be replaced.

QUIET

For file copy and remap, no output report is generated so QUIET is redundant.

For library copy or remap, on successful completion of a copy of one or more library members, QUIET suppresses display of the IEBCOPY execution report or the FCOPY PDS Copy Statistics list window. In this case, only FCOPY messages will be displayed.
Default is to display any IEBCOPY report or FCOPY statistics.

NEW

Applicable to library copy or remap only, indicates that, if the *to_fileid* library is not yet unallocated, then it should be allocated as new using the same DCB geometry and SPACE attributes as the *from_fileid* library.
NEW is ignored for file copy.

If this option is not selected, then the library will not be created and ZZSD353E open error message is returned.

Note that NEW is not a valid option if FCOPY is executed in batch.

Examples:

```
FCOPY   CBL.SEV.X628263.REPORT
```

Open the File Copy dialog window and populate the "From DSN" field with CBL.SEV.X628263.REPORT.

```
FCOPY   NBJ.JCL(ZZS* ZZI*)    NBJ.COPY.JCL    NEW
```

Create a copy of a JCL library containing only members with names starting "ZZS" or "ZZI".

```
FCOPY   NBJ.JCL(ZZS* ZZI*)    NBJ.COPY.JCL    REPLACE
```

Following execution of the previous example, take up-to-date copies and add any additional members with names starting "ZZS" or "ZZI".

```
FCOPY  /u/smpnts/X0000012/S0005.CBL.PROD.SERVICE.SVCRNTS   NBJ.JCL(SVCRNTS)  JCL
```

Create a batch job to copy an HFS file to a PDS library member.

FORWARD

Syntax:

```
>>--+- FOrw -----+---------------------------------------------------------><
    |            |
    +- FORWARD --+
```

Description:

Use the FORWARD command to scroll the window contents forwards by a page.

Note that FORWARD is the same as DOWN CURSOR where the cursor is outside the display area.

FS

Syntax:

```
>>--+- FS ---------+--+------------------------+----------------------------><
    |              |  |                        |
    +- FILESEARCH -+  +-- filemask -- string --+
```

Description:

Use the FS command to open the File Search Window and optionally perform a file search.

The File Search window may also be opened via the Utilities menu of the SELCOPY/i main window menu bar.

Parameters:

filemask

The file mask of the MVS PDS and member, the VSE LIBR lib.sublib and member or the CMS file name type and mode to be searched.

This parameter is placed in the Dataset field of the File Search window.

string

The search string.

This parameter is placed in the Search String field of the File Search window.

Examples:

`FS`

Open the File Search window with both the Dataset and Search String fields left blank.

`FS  'CBL.Q6930.JCL(VV*)'   'PGM=CBLV'`

Search PDS members beginning 'VV' for string 'PGM=CBLV'.

`FS  'PRD2.CBL.*.htmlL'      '<H1>'`

Search VSE PRD2.CBL library members of type 'HTML' for string '<H1>'.

`FS  '* EXEC A'              'SELCOPY'`

Search CMS EXEC files on the A minidisk for string 'SELCOPY'.

FSU

Syntax:

```
>>- FSU -+--------------------------------------------------------------+----><
         |                                                              |
         +-+-| Unformatted File Search Opts |--+----| Common Opts |-----+
           |                                   |
           +-| Unformatted File Update Opts |--+
           |                                   |
           +-| Unformatted File Copy Opts |----+
           |                                   |
           +-| Unformatted Library Copy Opts |-+
           |                                   |
           +-| Formatted File Search Opts |----+
           |                                   |
           +-| Formatted File Update Opts |----+
           |                                   |
           +-| Formatted File Copy Opts |------+
           |                                   |
           +-| Formatted File Remap Opts |-----+
           |                                   |
           +-| Formatted Library Copy Opts |---+
           |                                   |
           +-| Formatted Library Remap Opts |--+
```

Unformatted File Search Opts:

```
 >---| File Input |------------------| Search Opts |------------------------->
```

Unformatted File Update Opts:

```
 >---| File Input |----------------+-----------------+--| Change Opts |------>
                                   |                 |
                                   +-| Search Opts |-+

      +-- NOUpdate --+
      |              |
 >----+--------------+------------------------------------------------------->
      |              |
      +-- UPDate ----+
```

Unformatted File Copy Opts:

```
 >---| File Input |----------------+-------------------------------------+--->
                                   |                                     |
                                   +-----------------+--| Change Opts |--+
                                   |                 |
                                   +-| Search Opts |-+

 >--- OUTPUT fileid ----+----------+----------------------------------------->
                        |          |
                        +- APPend -+
```

Unformatted Library Copy Opts:

```
 >--| Library Input |----------------+-------------------------------------+->
                                     |                                     |
                                     +-----------------+--| Change Opts |--+
                                     |                 |
                                     +-| Search Opts |-+

 >--- OUTPUT lib_dsn ----+-------+--+-----------+---------------------------->
                         |       |  |           |
                         +- NEW -+  +- REPlace -+
```

Formatted File Search Opts:

```
>---| File Input |--| SDE Opts |----| Search Opts |------------------------->
```

Formatted File Update Opts:

```
>---| File Input |--| SDE Opts |--+-----------------+--| Change Opts |------>
                                  |                 |
                                  +-| Search Opts |-+

     +-- NOUpdate --+
     |              |
>----+--------------+------------------------------------------------------->
     |              |
     +-- UPDate ----+
```

Formatted File Copy Opts:

```
>---| File Input |--| SDE Opts |--+-------------------------------------+--->
                                  |                                     |
                                  +-----------------+--| Change Opts |--+
                                  |                 |
                                  +-| Search Opts |-+

>--- OUTPUT fileid ----+----------+----------------------------------------->
                       |          |
                       +- APPend -+
```

Formatted File Remap Opts:

```
>---| File Input |--| SDE Opts |--+-------------------------------------+--->
                                  |                                     |
                                  +-----------------+--| Change Opts |--+
                                  |                 |
                                  +-| Search Opts |-+

                             +- SDO ---+
                             |         |
>--- OUTPUT fileid -- USING -+---------+-- out_struct ------+----------+---->
                             |         |                    |          |
                             +- COBOL -+                    +- APPend -+
                             +- PL1 ---+
                             +- ADAta -+
```

Formatted Library Copy Opts:

```
>--| Library Input |--| SDE Opts |--+-------------------------------------+->
                                    |                                     |
                                    +-----------------+--| Change Opts |--+
                                    |                 |
                                    +-| Search Opts |-+

>--- OUTPUT lib_dsn ----+-------+--+-----------+---------------------------->
                        |       |  |           |
                        +- NEW -+  +- REPlace -+
```

Formatted Library Remap Opts:

```
>--| Library Input |--| SDE Opts |--+-------------------------------------+->
                                    |                                     |
                                    +-----------------+--| Change Opts |--+
                                    |                 |
                                    +-| Search Opts |-+

                             +- SDO ---+
                             |         |
>-- OUTPUT lib_dsn -- USING -+---------+-- out_struct ---------------------->
                             |         |
                             +- COBOL -+
                             +- PL1 ---+
                             +- ADAta -+

>-----------------------+-------+--+-----------+---------------------------->
                        |       |  |           |
                        +- NEW -+  +- REPlace -+
```

File Input:

```
             +---------------+
             v               |
>-- INPut ( --- fileid_mask -+-) --+--------------+------------------------->
                                   |              |
                                   +-| HFS Opts |-+
```

HFS Opts:

```
              +-- STD -----+
              |            |
   +-- EOL ---+------------+---------------+
   |          |            |               |
   |          +-- CR ------+               |
   |          +-- LF ------+               |
   |          +-- NL ------+               |
   |          +-- CRLF ----+               |
   |          +-- LFCR ----+               |
   |          +-- CRNL ----+               |
   |          +-- string --+               |  +- LRECL lrecl -+
   |                                       |  |               |
>--+---------------------------------------+--+---------------+------------->
   |                                       |
   +-- RECFM -+- F ------------------------+
              |                            |
              |     +- (0,2,0) ----------+ |
              |     |                    | |
              +- V -+- (off,len,origin) -+-+

>--+-----------+-+-----------+---------------------------------------------->
   |           | |           |
   +- RECURSE -+ +- CASEIgn -+
```

Search Opts:

```
                                                        (1)
                                            +------- AND -------+
                                            V                   |
>---+--+----------------------+--- Find (-+--- ( find_parms ) --+-+-) --+--->
    |  |                      |           |                       |     |
    |  +- WHere expression ---+           |            (2)        |     |
    |                                     | +------- OR --------+ |     |
    |                                     | V                   | |     |
    |                                     +--- ( find_parms ) --+-+     |
    |                                                                   |
    +---- WHere expression ---------------------------------------------+
```

Change Opts:

```
                             (1)
                 +------- AND -------+
                 V                   |
>---- Change (-+-- ( change_parms ) -+-+-) --------------------------------->
               |                       |
               |            (2)        |
               | +------- OR --------+ |
               | V                   | |
               +-- ( change_parms ) -+-+
```

Library Input:

```
             +------------+
             v            |
>-- INPut ( --- lib_mask -+-) ---------------------------------------------->
```

SDE Opts:

```
          +- SDO ---+
          |         |
>- USING -+---------+- in_struct -+---------------+-+---------------------+->
          |         |             |               | |                     |
          +- COBOL -+             + VIEW rectype -+ |         +--- , ---+ |
          +- PL1 ---+                               |         v         | |
          +- ADAta -+                               + SELect --+ field -+-+
                                                               |          |
                                                               +-- * -----+
```

Common Opts:

```
>--+----------------------------+-+--------------+--+---------------------+->
   |                            | |              |  |                     |
   +-+- FROM -----+- start_rec -+ +- FOR n_recs -+  +- REPORT -+- fileid -+
   | +- STARTREC -+             |                   +- RPT ----+          |
   |                            |                   |                     |
   +--- STARTKEY --- start_key -+                   +- NOREPORT ----------+
   |                            |                   +- NORPT -------------+
   +--- STARTRBA --- start_rba -+
```

Notes:

1. **AND** logical operator literal or its equivalent symbol: "&" (ampersand).
2. **OR** logical operator literal or its equivalent symbols: "¦" (broken bar) or "|" (vertical line).

Description:

The FSU command is the command line interface to the File Search/Update/Copy/Remap Utility.

If FSU is executed with no parameters, the File Search/Update/Copy/Remap Panel is opened.

Parameters specified on the FSU command govern the type of operation performed by the utility. The following table illustrates the operation performed when FSU parameters, denoted by "*" (asterisk), are provided.

	FIND	WHERE	CHANGE	USING	OUTPUT	OUTPUT USING
Unformatted Search	*	(1)	-	-	-	-
Unformatted Search	(1)	*	-	-	-	-
Unformatted Update	(1)	(1)	*	-	-	-
Unformatted Copy	(1)	(1)	(1)	-	*	-
Formatted Search	*	(1)	-	*	-	-
Formatted Search	(1)	*	-	*	-	-
Formatted Update	(1)	(1)	*	*	-	-
Formatted Copy	(1)	(1)	(1)	(2)	*	-
Formatted Remap	(1)	(1)	(1)	*	*	*

Notes:

1. At least one of FIND, WHERE, CHANGE or OUTPUT must be specified.
2. If the fileid specified on OUTPUT is a PDS/PDSE library DSN with no member name, then **Library copy/remap** is performed so that input library members are copied to members of the same name in the output library. Otherwise, File copy/remap occurs where all input records are copied to a single output file.

Unless restricted by STARTREC/STARTKEY/STARTRBA and/or FOR parameters, **all** records of a file (sequential, VSAM, PDS/PDSE and HFS) whose DSN/fileid matches the fileid mask(s) provided by the INPUT parameter, are included in the operation. If a structure, provided by the USING parameter, is applied to the input records, then only those records assigned the default record type (RTO), specified by the VIEW parameter, are eligible for search and change operations.

Any section or sections of the FSU command stream may be commented out using REXX style comment delimiters. (i.e. enclose areas of the command stream text between "/*" and "*/".) This is particularly useful when FSU commands are entered in the user's

HOME (CMX) command centre file, where the command may span a number of (continued) lines. e.g. To temporarily omit the CHANGE parameter, thus allowing the user to identify those records that would be selected for change...

```
<sd fsu                                               \
   INPUT (                                            \
           CBL*:JGE*.CBLI.SDE.SAMP.VAR(DATS*)         \
              %user%.CBLI.SDE.SAMP.VAR.DATS*.ESDS     \
         )                                            \
   WHERE (    CUST-ID > 5000                          \
         )                                            \
/*                                                    \
   CHANGE( (c'Aqua' c'AQUA'   (COMPANY)     )   AND   \
           (c'Jim'  c'James' (NAME,#9:#11) )          \
         )                                            \
      NOUPDATE                                        \
*/                                                    \
   USING      %user%.CBLI.SDO(CobSALES)               \
   VIEW       REC-CARD
```

The FSU command may be executed in the foreground or via SDEIN input to program SDEAMAIN for batch processing.

During foreground execution, the FSU output report is displayed and automatically updated in an SDE window view. A progress window is also displayed which allows the user to interrupt processing before completion using the Attention key. See File Search/Update/Copy/Remap Output.

Parameters:

`INPUT (`*`fileid_mask`*`< ...> | `*`lib_mask`*`< ...>)`

Specifies one or more file masks, *fileid_mask*, used to identify files to be searched, updated, copied or remapped. Multiple instances of *fileid_mask* must be specified with one or more intervening blanks.

All HFS files; cataloged sequential, GDG or VSAM data sets; cataloged PDS/PDSE libraries and library members whose names match a *fileid_mask*, will be selected for processing.

If, however, *fileid_mask* includes a volume mask, then both cataloged and uncataloged data sets whose names match the *fileid_mask* **and** reside on the specified volume will be selected for processing.

lib_mask is a sub-category of *fileid_mask* and may be specified as a PDS/PDSE library DSN mask, with or without an accompanying member mask, or one or more PDS/PDSE libraries allocated to a DDname. A differentiation is made between *lib_mask* and *fileid_mask* when performing **Library Copy**/**Remap** where only input PDS/PDSE libraries are processed. Any other file type selected for input to Library Copy/Remap will be not be copied.

A single *fileid_mask* may be in one of the following formats:

1. A pre-allocated DDNAME (non-HFS) which may represent one or more (concatenated) data sets and/or libraries. (e.g. SYSEXEC)

2. An absolute or relative HFS path name.

   Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in the name portion of the HFS path. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.

   Beacuse FSU supports comment specification (text enclosed between "/*" and "*/"). Where the HFS file name wild card "*" (asterisk) is to be used following a directory separator "/" (slash), the HFS path must me enclosed in single quotes (apostrophes) or double quotes. e.g.

   ```
   '/u/ibmuser/tmp/*'    /* Search all files in this directory. */
   ```

3. A DSN Mask and optionally a volume Mask and/or multiple PDS/PDSE member masks in the following format:

   ```
   {volmask:}data.set.name.mask{(membmask{,membmask, ...} )}
   ```

   fileid_mask must **not** be enclosed in quotes (TSO prefix is not used.)

   Wild card characters "%" (percent), "*" (asterisk) and "**" (double asterisk) are supported. (See File Search/Update/Copy/Remap Panel documentation for use of wild cards in the volume, DSN and member masks.) Similarly, one or more member masks may be specified between a single pair of "( )" (parentheses). Multiple PDS/PDSE member masks must be separated by a "," (commma) and/or one or more intervening blanks.
   e.g. `DEV88%.CBL*.**(SELC*, *MAN, XM*J*)`

   All sequential, VSAM and PDS/PDSE data sets that match a *fileid_mask* are selected for input. If one of these data sets is a PDS/PDSE library then all members of that library will be processed. In order to restrict the search to a single PDS/PDSE library and so exclude any non-PDS data set that matches *fileid_mask*, then a member mask should be specified.
   e.g. `SYS7.OEM.CBL202.CBLI.CBLE(*)`

   fileid_mask may be prefixed by a volume serial mask in order to restrict the search to only those cataloged and uncataloged data sets that match the specified *fileid_mask* **and** for which extents exist on the specified volume(s). The volume serial mask must be destinguishable from the rest of the fileid mask by an intervening ":"

(colon) with no embedded blanks.
e.g. `CBLM04:SYS7.**.DZ3*.**`

Note that, if this criteria is satisfied, then **all** records of a **cataloged**, multi-volume data set will be searched. However, only extents that exist on the specified volumes will be searched if the multi-volume data set is **uncataloged**.

HFS Opts

Applicable to **all** HFS files that match the specified *fileid_mask*, the following options may be specified to determine how HFS data is processed by the utility.

For non-HFS files that match an INPUT *fileid_mask*, HFS options are ignored.

`EOL=STD|NL|CR|LF|CRLF|LFCR|CRNL|`*`string`*

Specify the EOLIN (input end-of-line) delimiter used to identify the end of each record for unformatted (non-RECFM F or V) HFS file input. EOLIN delimiters are not included in the edited record data or record length. EOL parameter elements are as follow:

STD	-	Any standard line-end.
NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
string	-	A 2-byte user specified character or hex string.

STD is default so that the EDIT operation scans the input data for any of the standard EOL combinations (not *string*), stopping when one is found. This EOL combination is used as EOLIN for the file.

`RECFM F | V (`*`off,len,origin`*`)`

Specifies that the data is to be treated as containing Fixed or Variable length format records.

RECFM F indicates that all records are of a fixed length as defined by the LRECL argument.

RECFM V allows the user to specify the location of the record length fields within the data as follows:

off	Offset of the record length field from the start of the record.
len	Length of the record length field.
origin	The start of the record data at which the record length is applied.

Default is (0,2,0) which describes standard RECFM V organisation data sets.
The length field will be included as part of the input record data, so, if a CHANGE operation is specified, care must be taken not to corrupt the length field.

`LRECL `*`lrecl`*

Specifies the maximum record length of input HFS file records.

For RECFM F HFS files, *lrecl* is the fixed length of the records processed. If the HFS file size is not a multiple of *lrecl* value, then the last record of the file will be short.
Default *lrecl* for this types of file is 80.

For RECFM V and unformatted (EOL delimitted records) HFS files, if a record length exceeds *lrecl*, processing is stopped for that particular file.
Default *lrecl* for these types of files is 32752.

`RECURSE`

For an HFS *fileid_mask* containing wild card characters, recursively search files within all directories and sub-directories identified by the mask.
Default is not to search files in HFS sub-directories.

`CASEIGN`

Bypass case sensitivity for the **name** portion of all specified HFS path fileid masks. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.
Default is to respect the character case of the HFS file name.

SDE Opts

Specification of SDE parameters are required for, and applicable only to, **Formatted File or Library Search**/**Update**/**Copy**/**Remap** processing.

If SDE option parameters are not specified, **Unformatted File or Library Search**/**Update**/**Copy** processing is performed.

`USING SDO|COBOL|PL1|ADATA `*`in_struct`*

Specifies *in_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map input record data fields for use in a Formatted File Search, Formatted File Update, Formatted File Copy or Formatted File Remap operation.

All input records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records, and fields for WHERE, FIND and CHANGE operations.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. This is an overhead which may be overcome by generating a non-temporary SELCOPY/i SDO file and referencing that instead. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO) defined in the specified or generated SDO and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

SDO, COBOL, PL1 or ADATA identifies the source format of the input structure file.
Default is SDO.

SDO	A SELCOPY/i Structured Data Object. This is the format required by SELCOPY/i to format structured data.
COBOL	A COBOL copy book. SELCOPY/i will use the COBOL compiler to compile the file in order to generate a temporary SDO.
PL1	A PL1 include file. SELCOPY/i will use the PL1 compiler to compile the file in order to generate a temporary SDO.
ADATA	An ADATA file created by a previous COBOL or PL1 compilation of a copybook/include file. SELCOPY/i will use the ADATA file to generate a temporary SDO.

`VIEW` *`rectype`*

Applicable to **formatted** records only, *rectype* specifies the default record type against which all SELECT, WHERE, FIND and CHANGE operations are performed. Records not assigned this record type are not processed by these operations.

The named record type must match one defined within the specified SDO structure or within a temporary SDO structure generated from a COBOL, PL1, ADATA file referenced by USING.

Although not mandatory, it is recommended that VIEW *rectype* is included when performing formatted data processing so avoiding confusion over which records are to be processed. This is particularly true when performing CHANGE operations.
Default is the first record type defined in the structure (SDO).

`SELECT` *`field<, ...><, *>`*

Applicable to **formatted** records only, *field* identifies an individual field (by name or reference number) in records assigned the default record type, to be selected for FIND and/or CHANGE processing.

Multiple, comma separated *field* arguments may be specified, not only defining a list of fields, but also the order in which they are to be processed. "*" (asterisk) may be specified to represent all remaining fields in the default record type that have so far not been selected.

This is most useful where FIND and CHANGE operations are allowed to default to searching all fields. If CHANGE and/or FIND operations are specified to search specific fields that are not identified by SELECT, then processing stops and error ZZSD179E is returned.

Default is to search all fields in the record in their order of occurrence within the record type definition.

`WHERE` *`expression`*

Specifies an SDE expression to be applied by an SDE WHERE record filtering operation.

The WHERE expression defines search criteria for Unformatted File Search or Formatted File Search, or is used for record filtering before executing a FIND or CHANGE operation.

For **formatted** records, the WHERE operation is performed only on those input records that are assigned the default record type as specified by VIEW. Any field within the formatted record may be referenced regardless of whether it has been included by the SELECT.
e.g.

```
 FSU INPUT( DEV.USER01.JCL(*) )
     WHERE( (#3 >= 22)  AND  ((EmpName = 'Smith') OR (Dept >> 'E1')) )
```

For **unformatted** records, the WHERE operation is performed on all records. The WHERE expression may only include reference to a single field (field reference #1, field name "Record") which evaluates to all data in the focus record. This field has a data type of CHAR and length equal to the file's maximum record length.
e.g.

```
 FSU INPUT( DEV.USER01.JCL(*) )    WHERE( #1 >> '//'  AND  #1 << 'EXEC' )
```

If no CHANGE operation is specified, then a simple file search is performed so that the report identifies all records that satisfy the WHERE operation and any subsequently executed FIND operation. If a CHANGE operation is specified, only these records are eligible to be changed and only records that have been changed are identified in the output report.

e.g.

```
FSU INPUT( DEV.TEST.DATAX3. )  USING( DEV.TEST.SDO(SDDATAX3) )
 WHERE( JOBTITLE = 'MANAGER' OR SALARY > 38000 )
 CHANGE( ('Jo W Smith' c'JWS' (#20:#22)) OR ('J W Smith' c'JWS' (#20:#22)) )
```

FIND (*find_parms*) | ((*find_parms*) *op* (*find_parms*) <*op* ...>)

Specifies *find_parms* which corresponds to a search string and other supported parameters to be executed by an SDE FIND operation.

A FIND *find_parms* operation defines a search criterion for Unformatted File Search or Formatted File Search, or is used for record filtering before executing a CHANGE operation.

The FIND operation is performed on those input records that first satisfy any supplied WHERE expression.

Unlike SDE file edit, *find_parms* are applied at the record level, not at the file level. Therefore, *find_parms* parameters ALL/NEXT/FIRST/PREV/LAST all have the same effect and so are redundant.

Furthermore, for **formatted** records, only records assigned a record type that matches the VIEW default record type are searched. The FIND operation may be further restricted to search only selected input fields as specified by the SELECT parameter. If *find_parms* includes field references and SELECT is specified, then all fields identified by *find_parms* must also be referenced by SELECT, otherwise processing stops and error ZZSD179E is returned.

For **unformatted** records, if no WHERE expression is specified, FIND will be performed on all input records.

If multiple *find_parms* combinations are specified, each *find_parms* group must be enclosed in "( )" (parentheses) with intervening operator *op*, logical AND or logical OR. Note that a combination of both AND and OR logical operators is invalid. This indicates whether a record must satisfy all *find_parms* operations or only one of the *find_parms* operations in order to be selected.
e.g.

```
FSU INPUT( DEV.USER01.JCL(*) )
     FIND( (c'EXEC' WORD 1 20)  AND (c'IKJEFT01' NEXT)  AND ('REGION' NEXT) )

FSU INPUT( DEV.USER*.COBOL.COPYBOOK(*) )
     FIND( (c'REDEFINES' WORD)  OR ('OCCURS' WORD)  OR ('filler.' 12 80) )

FSU INPUT( DEV.TEST.DATAX3. )
    USING( DEV.TEST.SDO(SDDATAX3) )
     FIND( (25 (#10:#18)) OR ('Ramsay' (EMP_NAME)) )
```

Where logical AND is used, each execution of a FIND operation will perform a scan for the search string within the **entire width** of the record data and is not subject to the position of the data found by a previous, successful FIND *find_parms* operation.

For formatted records, a numeric search string will be treated as a signed numeric value and an arithmetic compare will occur for numeric data fields. For non-numeric fields and unformatted records, all search strings are treated as character data and a logical string compare is performed.

CHANGE (*change_parms*) | ((*change_parms*) <*op* (*change_parms*) <*op* ...>)

Specifies *change_parms* which corresponds to a search and replace string and other supported parameters to be executed by an SDE CHANGE operation.

CHANGE *change_parms* will perform character string or numeric value substitution on record data and, where OUTPUT has **not** been specified, implies Unformatted File Update or Formatted File Update.

The CHANGE operation is performed on those input records that first satisfy any supplied (WHERE and/or FIND) search criteria.

Furthermore, for **formatted** records, only records assigned a record type that matches the VIEW default record type are processed by CHANGE. The CHANGE operation may be further restricted to change only data in selected input fields as specified by the SELECT parameter. If *change_parms* includes field references and SELECT is specified, then all fields identified by *change_parms* must also be referenced by SELECT, otherwise processing stops and error ZZSD179E is returned.

For **unformatted** records, if no FIND or WHERE parameter is specified, CHANGE will be performed on all input records.

Unlike SDE file edit, *change_parms* are applied at the record level, not at the file level. Therefore, if specified, *change_parms* parameter ALL changes all occurrences of the CHNAGE search string within a record, FIRST or NEXT change the first occurrence and LAST or PREV change the last occurrence within a record.

If multiple *change_parms* combinations are specified, each *change_parms* group must be enclosed in "( )" (parentheses) with intervening operator *op*, logical AND or logical OR. Note that a combination of both AND and OR logical operators is invalid. If logical operator AND is used, CHANGE will execute all the *change_parms* operations. If logical operator OR is used, CHANGE will execute each *change_parms* operation in turn until one successfully changes the data, at which point, no further *change_parms* operation is attempted.
e.g.

```
FSU INPUT(OEM.**)
   CHANGE( (c'DB8F' c'DB9G' PREFIX ALL) AND (c'CBLI' c'CBLI160' ALL) )

FSU INPUT(DEV.USER01.JCL(*) )
```

```
      FIND(c'EXEC' WORD 1 20)
    CHANGE(c'IEWL' c'BIND' WORD FIRST 16 80)

 FSU INPUT(DEV.TEST.DATAX3)
     USING(DEV.SDO(DATAX3))
    CHANGE(('Jo Smith' c'JS' (#2:#5)) OR ('J W Smith' c'JS' (#2:#5)))
```

Where logical AND is used, a change made by a *change_parms* specification may itself be changed by a subsequent *change_parms* specification within the same execution of FSU. Also, each execution of a CHANGE operation will perform a scan for the search string within the **entire width** of selected record data and is not subject to the position of the data found (or changed) by a previous, successful FIND *find_parms* or CHANGE *change_parms* operation.

For Formatted records, a numeric search string and replace string will be treated as a signed numeric values. An arithmetic compare will occur for the search string when applied to numeric data fields and the numeric replace string converted to a field's numeric data type as appropriate. For non-numeric fields and unformatted records, all search and replace strings are treated as character data and a logical string compare is performed.

Note that, for File Update only, records are re-written using update-in-place so the record length cannot be changed. Therefore, the CHANGE operation must not alter the length of an unexpanded/unformatted record, otherwise a change error will occur. This condition will be flagged against the record in the output report.

Where the length of a search string is different to that of the replace string, then the following occurs:

- ♦ If the length of search string is greater than the length of the replace string, then words to the right of the replaced string will be shifted left.

  However, if parameter TEXT is specified and more than one blank exists before a word to the right of the replaced string, then blanks are inserted to maintain that word's position in the record.

- ♦ If the length of the search string is less than the length of replace string, then words to the right of the replaced string will be shifted right. Note, however, that CHANGE will not increase the length of formatted data beyond its defined maximum field length.

  If parameter TEXT is specified, multiple, consecutive blanks are absorbed to leave at least one blank between each word. Only if no blanks are eligible to be absorbed will text to the right of the replaced string be shifted right.

```
NOUPDATE
UPDATE
```

NOUPDATE and UPDATE are applicable to Unformatted File Update and Formatted File Update only.

NOUPDATE indicates that records that would be updated by the CHANGE operation are not to be written to disk so allowing the user to first review the output report and verify that the changes are correct before re-running the FSU command with UPDATE.

UPDATE indicates that records altered by the CHANGE operation are written to disk, so replacing the previous copy of the record.

Default is NOUPDATE.

Copy/Remap Opts

Copy/Remap parameters are required for, and applicable only to, **Copy** and/or **Remap** processing.

Specification of OUTPUT defines the utility processing to be Unformatted File or Library Copy or Formatted File or Library Copy. If an Output USING structure is also specified, then processing will be Formatted File or Library Remap.

If Copy/Remap parameters are not specified, File **Search** or **Update** processing is performed.

`OUTPUT fileid | lib_dsn`

Specifies *fileid* or *lib_dsn*, an output file or library to which **all** input records will be copied.

lib_dsn is a new or existing PDS/PDSE library **without** a member name specification. If specified, *lib_dsn* indicates that **Library Copy/Remap** will occur so that input library members are copied to *lib_dsn* with their member name unchanged. Non-library member input files are ignored.

fileid may be one of the following:

- An existing physical sequential (PS) data set.
- An existing VSAM data set.
- A new or existing member of an existing PDS/PDSE library.
- A new or existing HFS/ZFS file path.

If a CHANGE operation is specified and activated, then record data may be changed before it is written to the output file.

The format of the output file should be compatible with input record data. e.g.

- If an output KSDS data set is specified, input records must be in key sequence, as defined by the output file, and must not contain duplicate keys. If an input record does not satisfy these conditions, it will fail to copy.

- Records will be truncated if the input record length exceeds the maximum allowed by the output file.

`USING SDO|COBOL|PL1|ADATA out_struct`

Specifies *out_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map output record data fields for use in Formatted File or Library Remap.

Formatted File Remap processing only occurs when an output file and input and output structures are supplied. Therefore, an output structure is ignored if no input structure has been specified.

During the remap process, the following will occur:

1. Input records of record type not defined in the output structure are copied without field remap.
2. Output structure record types not defined in the input structure are redundant and so are ignored.
3. Record data in input fields are copied to output fields of the same name belonging to record types of the same name.
4. The input field data will be reformatted to the data type of the output field and will be moved to the output field's position within the record map.
5. Any input fields whose field names are not part of the output record structure, will not be included in the output record.
6. Any output fields whose field names are not part of the input record structure are initialised to their default values.

See the input structure USING field for description of output USING field sub-parameters and implementation of a structure on record data.

`APPEND`

Applicable to File copy or remap only, specifies that output records are to be appended to existing data in the output file.

If this option is not selected, then the existing records will be overwritten.

`NEW`

Applicable to Library copy or remap only, indicates that, if *lib_dsn* is as yet unallocated, then the library should be allocated as new using the same DCB geometry and SPACE attributes as the first input library identified by the INPUT *lib_mask*.

If this option is not selected, then the library will not be created and ZZSD353E open error message is returned.

`REPLACE`

Applicable to Library copy or remap only, indicates that, members that exist in *lib_dsn* will be replaced by input members of the same name. Note that, if more than one input library contains a member of the same name, then both will be copied but the second member copied will replace the first.

If this option is not selected, then existing members will not be replaced.

`STARTREC start_rec`
`FROM`

STARTREC (or FROM) specifies the record number *start_rec* of the first record to be processed in all input files identified by INPUT *fileid_mask* or *lib_mask*. All records occurring before *start_rec* are bypassed.

start_rec may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

If STARTREC is not specified, *start_rec* defaults to 1.

`STARTKEY start_key`

For VSAM KSDS, VRDS files or PATHs only, STARTKEY specifies a full or partial key *start_key* used to identify the first record to be processed in all input files identified by INPUT *fileid_mask* or *lib_mask*. All records occurring before *start_key* are bypassed.

start_key may be specified as a character or hex string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing of *start_key* will occur if specified as a character string without the "C" (or "c") prefix.

The record selected by *start_key* will be the first record with key field data which is greater than or equal to *start_key*.

`STARTRBA start_rba`

For VSAM ESDS files only, STARTRBA specifies a relative byte address *start_rba* used to identify the first record to be processed in all input files identified by INPUT *fileid_mask* or *lib_mask*. All records occurring before *start_rba* are bypassed.

start_rba may be specified as a decimal integer or hexadecimal value.

The record selected by *start_rba* will be the first record with a relative byte address which is greater than or equal to *start_rba*.

`FOR n_recs`

FOR specifies the maximum number of records *n_recs* to be processed from each input file identified by INPUT *fileid_mask* or *lib_mask*.

If FOR is not specified, *n_recs* is unlimited.

`REPORT fileid`
`RPT`

Specifies *fileid*, the DSN of an existing sequential (PS) data set to which the search/update/copy/remap report records are to be written. The dataset name must be fully qualified, quotes being unnecessary but permitted.

The report is a structured data file designed to be browsed (not printed) using an SDE structure definition object (SDO), which will be generated automatically.

The associated SDO fileid is constructed simply by adding **.SDO** to *fileid*. Therefore, the DSN of the report file is restricted to 40 bytes in length.
Report output to an HFS dataset is not currently supported.

If this option is not specified, *fileid* defaults to "*user*.FSU.Dyyyyddd.Thhmmss" with SDO fileid "*user*.FSU.Dyyyyddd.Thhmmss.SDO".

```
NOREPORT
NORPT
```

Indicates that report generation is to be suppressed. This is most useful for copy processing without CHANGE, whereby no FSU report records are generated for input data records.

Examples:

Use of the FSU command may result in long command streams. Therefore, it is recommended that any FSU command should be entered as text in your HOME command centre (CMX) data set.

```
<sdata fsu input(XRVHC.**.PROCLIB(*) SYS1.PROCLIB(*))   find( DSN710 )
```

Report any member records within the specified PROCLIB libraries that contain the string "DSN710".

```
<sdata fsu input(XRVHC.**.PROCLIB(*) SYS1.PROCLIB(*)) change(DSN710 DSN810 ALL)
```

For member records within the specified PROCLIB libraries, report records that contain the string "DSN710" followed by the records' appearance after replacing all occurrences of "DSN710" to "DSN810". Members records are **not** updated.

```
<sdata fsu                                                              \
   INPUT ( SAR22.TEST.FX*.** )                                          \
   USING ( SAR22.FX100.COBOL.COPYBK.SDO )                               \
   VIEW  ( FX_Part_02 )                                                 \
   SELECT( Part_ID, Serial_No, Batch_No, Part_Description )             \
   WHERE ( Batch_No > 730   AND  (Fault_Type >> 'RTB'  OR  Quantity < 200) ) \
   FIND  ( c'Nut' PREFIX (Part_Description) )                           \
   CHANGE( 'screw' 'bolt' WORD (Part_Description) )                     \
   NOUPDATE
```

A Formatted File Update. Records from data sets and members of PDS/PDSE libraries whose DSNs match the specified fileid mask are filtered so that only records that are of the record type "FX_Part_02" and match the FIND and WHERE criteria are processed by the CHANGE operation. SELECT indicates a subset of fields eligible to be searched, updated, and displayed in the output report. Both the FIND and CHANGE arguments further restrict string location/update to text within the field "Part_Description" only.

FSUEND

Syntax:

```
>>-- FSUEND------------------------------------------------------------------><
```

Description:

Use the FSUEND command to save and close the display of report output generated by a foreground execution of the File Search/Update/Copy/Remap utility.

If the report and its associated SDO structure file has not been saved, then FSUOUT will prompt the user to save both these files before the report is closed. The report data will be saved as a VSAM ESDS data set and the SDO as a physical sequential data set. The user will be prompted to enter allocation values but the defaults are usually acceptable.

FSUEND is assigned to <PF3> by default when an IOError has been reported or when a File Update operation has been performed. Therefore, execution of FSUEND from a command prompt is only necessary if the report generated by a File Search, Copy or Remap is to be saved.

Parameters:

FSUEND has no parameters.

FSUOUT

Syntax:

```
>>-- FSUOUT----+-----------------------+-------------------------------------><
               |                       |
               +-- fsu_report_fileid --+
```

Description:

Use the FSUOUT command to display (browse) the saved report output from a previous execution of the File Search/Update/Copy/Remap utility.

FSUOUT executes the SDE EDIT command to display the specified FSU output report data set using a structure which has the same DSN but with additional low level qualifier ".SDO".

If no parameter is specified, the last saved FSU report output having the default report DSN format (*prefix*.FSU.Dyyyyddd.Thhmmss) is displayed.

FSUOUT may also invoked for an entry in a file list window using the "FO" prefix command.

Parameters:

fsu_report_fileid
The DSN of a report data set output generated by the File Search/Update/Copy/Remap utility.

Examples:

`FSUOUT   DEV.NBJ.FSU.D2008268.T114326`
Display the FSU report for DSN 'DEV.NBJ.FSU.D2008268.T114326' using an existing structure of DSN 'DEV.NBJ.FSU.D2008268.T114326.SDO'.

FSUUNDO

Syntax:

```
                                        +- Panel ------+  +- Verify --+
              (1)                       |              |  |           |
>>-- FSUUNDO --+---------------------+--+--------------+--+-----------+------>
               |                     |  |              |  |           |
               +- fsu_report_fileid -+  +- Foreground -+  +- Update --+
                                        |              |
                                        +- Background -+

                +- Terse ----+
                |            |
 >-------------+------------+--+--------------+---------------------------->< 
                |            |  |              |
                +- Extended -+  +- Diagnose ---+
```

Notes:

1. Parameters may be entered in any order.

Description:

Use the FSUUNDO command to execute the File Update Undo facility to restore all records in all files updated by the File Search/Update/Copy/Remap utility (FSU).

A File Search/Update/Copy/Remap report generated by the utility for an update operation, is used as input to FSUUNDO. This FSU report provides the affected fileids, the record numbers and images of the record data before and after the update was performed. Note that an attempt will be made to restore **all** records referenced as having been updated in the report.

Record data belonging to an entry flagged as "A" (After) in the "zT" field of the FSU report "Hit" records, will be replaced by data in the preceding FSU report record flagged as "B" (Before).

Before updating a record, the File Update Undo utility will first verify that the records data matches the updated record data in the report. This ensures that no restore will be performed if the record data has subsequent changes.

Unless parameter FOREGROUND or BACKGROUND is specified, FSUUNDO will open the File Search/Update/Copy/Remap panel. Other parameters specified on FSUUNDO will be reflected in the dialog fields.

Parameters:

fsu_report_fileid

Specifies the DSN of the File Search/Update/Copy/Remap Output report to be used to identify changed data set records.

The FSU report data set referenced may be that of **any** previous execution of the utility that involved a CHANGE operation.

Default is the last saved FSU report, or, optionally, the FSU report displayed in the current SDE edit window view. If the current SDE edit window view contains an FSU output report (identified by its DSN format), the user will be prompted to use this report instead.

PANEL
FOREGROUND
BACKGROUND

Specifies whether to open the FSUUNDO dialog window (PANEL), execute the FSUUNDO verify or update procedure immediately in TSO (FOREGROUND) or generate and display a JCL job deck suitable for submission to batch (BACKGROUND).

Default is PANEL.

VERIFY
UPDATE

Specifies whether to execute the UNDO procedure with or without performing an update of the record data.

VERIFY provides the user with the opportunity to execute a "dry run" to examine the FSUUNDO output report for any errors before proceeding with an execution for UPDATE. It is strongly recommended that FSUUNDO is executed with VERIFY prior to performing a run for UPDATE. Use of VERIFY will be indicated at the start and end of the FSUUNDO report with the additional record beginning "** Verify Only".

UPDATE will update records in the FSU reported data sets, so undoing the changes made by the File Update execution.

Default is VERIFY.

TERSE
EXTENDED

Specifies whether FSUUNDO is to output a brief (TERSE) or verbose (EXTENDED) report.

In a terse report output, data sets or PDS(E) members that have been updated without error are represented by a single report line and data sets that have already been updated by a previous FSUUNDO run are not reported. However, more detailed report output is generated if unexpected data is found and so an error condition flagged.

Extended report output will generate output for every successful or unsuccessful record update. See File Update Undo Output for more details.

Default is TERSE.

DIAGNOSE

Required only if a SELCOPY run time error occurs during execution of FSUUNDO, DIAGNOSE will remove the SELCOPY NOPRINT option and so write diagnostic report information to SYSPRINT.

If executing with parameter FOREGROUND, the SYSPRINT output is automatically displayed in a CBLe edit view with a DSN equal to the FSU output report DSN but with the additional low level qualifier "LIST". e.g. `NBJ2.DEV.FSU.D2008346.T162607.LIST`

HELP

Syntax:

```
>>-- Help --+-----------+---------------------------------------------------><
            |           |
            +-- topic --+
```

Description:

Use the HELP command to open the Help Window and optionally link directly to help on a specific CLI command.

Where topic is not specified or not found, the relevant table of contents is displayed.

The Help window may also be opened via the Help item of the window's menu bar.

Parameters:

topic

Display help on a specific topic.
If topic is enclosed in single or double quotes, the string is treated as the fileid of an HTML data set to be browsed. This may be the fully qualified fileid of an HTML document or the name of a PDS member that exists in the defualt HELP library.

Examples:

`HELP`

Open the Help window contents page.

`HELP CBLe`

Open the Help window at the CBLe command page.

`H "OEM.CBL.HTML(TEST)`

Open a specific HTML document library member.

`H "ZZSISIZE"`

Open the SELCOPY/i Help member name ZZSISIZE.

HOME

Syntax:

```
>>-- HOme -------------------------------------------------------------------><
```

Description:

Edit the user's personal command centre (CMX) file. A new CBLe text edit session is opened if one is not already open.

IEBCOPYDIALOG

Syntax:

```
>>---- IEBCOPYDialog --------------------------------------------------------><
```

Description:

The IEBCOPYDIALOG command may be used to open the Execute IEBCOPY dialog window to copy members between PDS(E) libraries.

The dialog window will be opened with fields populated with parameters entered by the user during the last invocation of the window.

Parameters:

IEBCOPYDIALOG has no parameters.

ISPF

Syntax:

```
>>--- ISPF --+----------------+----------------------------------------------><
             |                |
             +- ispf_command -+
```

Description:

When running in an ISPF environment, the SELCOPY/i command **ISPF** either toggles between using TSO and ISPF to manage 3270 I/O or executes an ISPF command. When used to execute an ISPF command, screen management is always handled by ISPF regardless of the current 3270 screen manager.

Note that when ISPF is the screen manager, the menu item **SwapList** is added to the CBLe main window menu bar. Selecting Swap will execute **ISPF SWAP LIST** to display ISPF's split screen menu.

It is recommended that, when running SELCOPY/i in an ISPF environment, ISPF should always be used as the 3270 screen manager to take advantage of ISPF screen split, etc. In order to do this without disrupting PFkey assignments, SELCOPY/i must run as an ISPF application with applid CBLI.

The *SELCOPY Product Suite Customisation Guide* provides instructions on customising SELCOPY/i to run as an ISPF application. When configured, there should never be any need to toggle back to TSO screen management. The supplied REXX macro, SELCOPYI, is used to run SELCOPY/i as an ISPF application with applid CBLI.

If SELCOPY/i is not defined as an ISPF application, then, when ISPF screen manager is used, SELCOPY/i function key definitions will be interpreted differently to those defined in SELCOPY/i. In this case, it is recommended that passing control to ISPF should only be carried out temporarily to perform ISPF explicit functions.

Toggling between ISPF and TSO screen management may also be achieved via the **Use TSO/ISPF** item of the System Menu.

Parameters:

ispf_command
ISPF command to be issued.

Examples:

`ISPF`
Set TSO as the screen manager if current screen management is done by ISPF **or** set ISPF as the screen manager if current screen management is done by TSO.

`ISPF SPLIT`
Set ISPF as the screen manager (if not already so) and execute ISPF SPLIT command so that the screen is split at the current cursor position.

ISPFUTIL

Syntax:

```
>>--+- ISPFUTIL -+-----------------------------------------------------------><
    |            |
    +- IU -------+
```

Description:

When running SELCOPY/i in ISPF, the SELCOPY/i command **ISPFUTIL** starts the ISPF Utility Selection Panel.

The ISPF panel is started as a full screen application which returns control to SELCOPY/i only after it is closed.

The ISPF Utility Selection Panel may also be started via the **Utilities** menu of the SELCOPY/i main window menu bar.

KEYS

Syntax:

```
>>--+- KEYS -----+--+-------------------------------------------------------+--><
    |            |  |                                                       |
    +- SETFKEYS -+  |  +-- DEFAULT ------+                                  |
    |            |  |  |                 |                                  |
    +- PF -------+  +--+-----------------+--+-------------------------------+
    |            |     |                 |  |                               |
    +- SFK ------+     +-- windowname ---+  +- pfn --+----------------------+
                       |                 |           |                      |
                       +-- windowclass --+           +------------+--  cmd -+
                       |                 |           |            |
                       +-- CAPTION ------+           +-- DELAY ---+
                       |                 |           |            |
                       +-- BORDER -------+           +-- BEFORE --+
```

Description:

Use the KEYS command to assign a command to a function key or display the Function Keys window.

Parameters:

null
Opens the Function Keys Command Tables for the current window.

DEFAULT
Selects the system default function key table.

CAPTION
Selects the window caption function key table.

BORDER
Selects the window border function key table.

windowname
Selects the function key table of the window name specified.

windowclass
Selects the function key table of the window class specified.

pfn
The function key number 1-24.

cmd
The command text string to be assigned. If null then the function key becomes unassigned.

DELAY
Place the function key command on the command line when the key is pressed rather than execute it.

BEFORE
Execute the function key command before processing any user screen inputs.

Examples:

```
KEYS  EDTWEDIT     16    'macro delblank'
```

Set PF16 to execute user edit macro DELBLANK for all windows of windowclass EDTWEDIT.

LA

Syntax:

```
>>--+- LA --------+--+-------------+------------------------------------------><
    |             |  |             |
    +- LISTALLOC -+  +-- ddname  --+
```

Description:

Use the LA (List Allocated files) command to open an Allocated Datasets List window and optionally list all MVS DD names or VSE file labels currently allocated to your job.

The Allocated Files window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS.

Parameters:

`ddname`

Select only list entries for the specified MVS DD name or VSE file label mask.

A *ddname* mask supports the following wild cards:

- `*` A single asterisk represents the complete MVS ddname/VSE label or zero or more characters within the *ddname* mask.
- `%` A single percent sign represents exactly one character within the *ddname* mask. Up to 8 percent signs can be specified in each *ddname* mask.

If no wildcards are specified within the *ddname* mask then wildcard "*" is appended to *ddname* so selecting all MVS ddnames/VSE labels that start with the specified *ddname*.

Examples:

`LA SYS`

List all allocated files beginning with 'SYS'.

LAS

Syntax:

```
>>--+- LAS ------------------+--+--------------------------------------------+--><
    |                        |  |                                          |
    +- LASSOC ---------------+  +-- entry --+---------------------------+--+
    |                        |              |                           |
    +- LISTASSOC ------------+              +-- catalog --+----------+--+
                                                          |         |
    +- LISTASSOCIATIONS -----+                            +- types -+
```

Description:

The LAS (List Associations) command may be used to open a Associations List window and optionally select cataloged entries for which associated objects will be displayed.

By default, the associated object name for a selected catalog entry is displayed in the first column. For example, if the selected entry is a VSAM CLUSTER object, a separate list entry may exist for its associated DATA, INDEX and ALTERNATE INDEX objects. Similarly, if the selected entry is a DATA object, a list entry will exist for the associated CLUSTER object but not an INDEX object which is associated with the CLUSTER, not the DATA object.

The Associations List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS or VSE.

Parameters:

`entry`

Specifies the fileid mask used to select cataloged entries, which is placed in the Entry field of the Associations List window.

The fileid mask represents a DSN mask that supports the following wild cards:

- `*` A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
- `**` A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
- `%` A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                becomes:   DEV*
DEV.OEM.TRSPAN*.     becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.       becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

   ```
   DEV                  becomes:   DEV.**
   DEV*                 becomes:   DEV*.**
   DEV.OEM.TRSPAN*      becomes:   DEV.OEM.TRSPAN*.**
   DEV.*.*SPA*          becomes:   DEV.*.*SPA*.**
   ```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

   ```
   DEV.OEM.TRSPAN       becomes:   DEV.OEM.TRSPAN*.**
   DEV.*.*SPA%          becomes:   DEV.*.*SPA%*.**
   SYS1.*.Z19           becomes:   SYS1.*.Z19*.**
   ```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

`catalog`

Specifies the catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> field.

An "*" (asterisk) may be specified to imply the default catalog name. This need only be specified if the *types* parameter is to be used.

Default is the master catalog.

The *catalog* string is placed in the Catalog field of the Catalog List window.

`types`

Specifies the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

Default is to select entries of all types.

The *types* string is placed in the Types field of the Catalog List window.

Examples:

```
lva   CBL.%%C
```
List associations for cataloged entries matching the fileid mask "CBL.%%C*.**".

```
lva   CBL.**   *    A
```
List associations for non-VSAM cataloged entries matching the fileid mask "CBL.**". (This will display defined ALIAS names for non-VSAM data sets.)

```
lva   NBJ      *    H
```
List associations for GDG data sets matching the fileid mask "NBJ*.**". (This will display the GDG Base name for each selected GDG data set.)

```
lva   NBJ.**.X232.   *    R
```
List associations for VSAM PATH entries matching the fileid mask "NBJ.**.X232".

LC

Syntax:

Open an MVS Cataloged Entries List Window:

```
>>--+- LC ----------+---+-------------------------------------------+--------><
    |               |   |                                           |
    +- LISTCAT -----+   +-- entry ----+--------------------------+--+
    |               |                 |                          |
    +- FL ----------+                 +-- catalog --+---------+--+
    |               |                               |         |
    +- FILELIST ----+                               +- types -+
```

Open a CMS File List Window:

```
>>--+- LC ----------+---+-------------------------------------------+--------><
    |               |   |                                           |
    +- LISTCAT -----+   +-- entry ----------------------------------+
    |               |
    +- FL ----------+
    |               |
    +- FILELIST ----+
    |               |
    +- LD ----------+
    |               |
    +- LISTDATASET -+
```

Open a VSE Catalog List Window:

```
>>--+- LC ----------+---+-------------------------------------------+--------><
    |               |   |                                           |
    +- LISTCAT -----+   +-- catalog --+--------------------------+--+
    |               |                 |                          |
    +- FL ----------+                 +-- entry ----+---------+--+
    |               |                               |         |
    +- FILELIST ----+                               +- types -+
```

Description:

For CMS, the LC command opens the File List window in place of the Catalog List or Dataset List window, and displays information about files residing on accessed mini-disks.

For MVS and VSE, the LC (List Catalog entries) command may be used to open a Catalog List window and optionally list basic information about entries in the catalog. This is a less detailed list than that generated by the LD command.

For VSE, the LC command is only supported where the CBL software product CBLVCAT is installed and active. The LC command uses CBLVCAT to read the specified VSAM catalog records and obtain information about the cataloged files.

The Catalog List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Parameters:

entry

Specifies the fileid mask which is placed in the Entry field of the MVS/VSE Catalog List window or the File field of the CMS File List window.

◊ For **MVS** systems, the fileid mask represents a DSN mask that supports the following wild cards:

`*` A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.

`**` A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.

`%` A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                 becomes:   DEV*
DEV.OEM.TRSPAN*.      becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.        becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                   becomes:   DEV.**
DEV*                  becomes:   DEV*.**
DEV.OEM.TRSPAN*       becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*           becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN        becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%           becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19            becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

◊ For **VSE** systems, the fileid mask is a valid CBLVCAT LISTCAT KEY parameter string. i.e. entries with file name **beginning** with the specified string or, if prefixed by "/" (slash), entries with file name **containing** the specified string. (See the CBLVCAT User Manual.)

If no fileid mask is specified, all entries will be selected.
Note that wild cards are not supported within the VSE fileid mask, however, "*" (asterisk) is tolerated if placed at the end of the fileid mask.

An "*" (asterisk) may also be specified in place of the fileid mask to imply that all entries are to be selected. This need only be specified if the *types* parameter is to be used.

For **CMS** systems, the fileid mask may consist of up to 3 qualifiers representing a filename filetype filemode combination where qualifiers are separated by one or more blanks or a "." (dot/period).

A single "*" (asterisk) wild card may be used to represent an entire qualifier or zero or more characters at a particular position within the qualifier. Wild card "*" may be specified more that once, anywhere within a qualifier.

Default CMS filemode qualifier is "A", default CMS filetype qualifier is "*".

`catalog`

Specifies the catalog in which to search for the requested entry.

For **MVS** systems, this is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> field.

An "*" (asterisk) may be specified to imply the default catalog name. This need only be specified if the *types* parameter is to be used.

For **VSE** systems, this is a disk label assigned to the VSAM catalog for which entries are to be listed.

Default for both MVS and VSE is the master catalog.

The *catalog* string is placed in the Catalog field of the Catalog List window.

`types`

Specifies the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

Default is to display entries of all types.

The *types* string is placed in the Types field of the Catalog List window.

See Also:

LD

Examples:

```
lc    CBL.%%C
```

List MVS cataloged entries matching the fileid mask "CBL.%%C*.**".

```
lc    CBL.SYS*.**    *    A
```

List MVS non-VSAM cataloged entries matching the fileid mask "CBL.SYS*.**".

```
lc    IJSYSCT   *    U
```

List user catalogs in the VSE VSAM master catalog.

```
lc    VSESPUC   /CICS
```

List all files containing the string "CICS" in the VSE VSAM catalog "VSESP.USER.CATALOG".

```
fl    CBL*.EXEC
```

List CMS files that match the fileid mask "CBL* EXEC A".

LD

Syntax:

```
>>-+- LD ----------+--+---------------------------------+--------------------><
   |               |  |                                 |
   +- LISTDATASET -+  +-- entry --+---------------------+
                                  |                     |
                                  +- catalog -+---------+
                                              |         |
                                              +- types -+
```

Description:

For CMS only, the LD and LC commands are synonyms for the FL command and so opens the File List window. Refer to documentation for LC (FL) for syntax and parameter specification.

The LD command is not supported on **VSE** systems.

For MVS, the LD (List Dataset details) command is used to open a Dataset List window and optionally list entries in the catalog together with the details of their geometry obtained either from the catalog or the VTOC. This is a more detailed list than that generated by the LC command.

Where *entry* is identified as being an HFS path (i.e. begins with "." or contains "/"), the HFS Path list window is opened instead.

The Dataset List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Parameters:

entry

Specifies the fileid mask used to identify required catalog entries.

The fileid mask represents a DSN mask that supports the following wild cards:

`*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.

`**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.

`%` A single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

Unless the **last** character of the fileid mask is a wild card "*" (asterisk) or a "." (dot/period), then a default trailing wild card string is appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last qualifier is length 8, a wildcard string of ".**" is appended. e.g.

```
DEV                     becomes:   DEV.**
DEV.OEM.TRSPAN00        becomes:   DEV.OEM.TRSPAN00.**
DEV.*.TRSPAN00          becomes:   DEV.*.TRSPAN00.**
```

2. A wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.CBL202         becomes:   DEV.OEM.CBL202*.**
SYS1.ZOS               becomes:   SYS1.ZOS*.**
```

The ammended *entry* string is placed in the Entry field of the Dataset List window.

`catalog`

Specifies the catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, and the required catalogs are searched.

An "*" (asterisk) may be specified to imply the default catalog name. This need only be specified if the *types* parameter is to be used. Default is the master catalog.

The *catalog* DSN searched is placed in the Catalog field of the Dataset List window.

`types`

Specify the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

The *types* parameter string is placed in the Catalog field of the Dataset List window.

See Also:

LC Command

Examples:

```
LD CBL.%%C
LD CBL.SYS*.**   USERCAT.CBLCAT  A
```

LEFT

Syntax:

```
>>-- LEFT --+--------------+--+-----------+---------------------------------><
            |              |  |           |
            +- windowname -+  +- CURSOR --+
                              |           |
                              +- DATA ----+
                              |           |
                              +- HALF ----+
                              |           |
                              +- MAX -----+
                              |           |
                              +- PAGE ----+
                              |           |
                              +- n_cols --+
```

Description:

Scroll the view of the data within the specified window left towards the first column of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_cols* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, LEFT, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to LEFT is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one column is used.

List windows may contain fields that have **KEY** attribute **YES** defined in the Field Descriptor Block. Fields with this attribute are always in view and may not be scrolled right or left. If the cursor is positioned in a column belonging to this type of field, then, for LEFT CURSOR and RIGHT CURSOR, the cursor is considered to be outside the display area. All columns of data that do not belong to a **KEY** field are scrollable using LEFT and RIGHT.

By default this command is assigned to function key **PF10**.

Parameters:

windowname
The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CURSOR
The scrollable column on which the cursor is positioned becomes the last scrollable column of the display.
If the cursor is positioned outside the display area, in a KEY field or on the last scrollable column within the display area, then LEFT PAGE is executed instead.

DATA
Scroll left so that the first scrollable column in the current display area becomes the last scrollable column of the display.

HALF
Scroll a number of columns so that the column situated half way along the width of the current display of scrollable columns, becomes the last scrollable column of the display.

MAX
Scroll left to display the first scrollable column of data.

PAGE
Scroll left so that the scrollable column of data to the left of the first scrollable column in the current display, becomes the last scrollable column of the display.

n_cols
Scroll left a specified number of floating columns.
The scrollable column of data that is *n_cols* to the left of the first scrollable column becomes the new first scrollable column of the display.

LJQ

Syntax:

```
>>--+- LJQ --------+--+-------------+---------------------------------------><
    |              |  |             |
    +- LISTJOBENQ -+  +-- jobname --+
```

Description:

Use the LJQ (List MVS Job Enqueues) command to open a Job Enqueue List window containing outstanding MVS enqueues held by a given job.

The Job Enqueue List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS or VSE.

Parameters:

jobname
The name of the job for which the ENQueues are to be listed.

This parameter is placed in the JobName field of the Job Enqueue List window.

See Also:

Examples:

```
LJQ  NBJTSO
```

List Enqueues for job NBJTSO.

LL

Syntax:

```
>>--+- LL ----------+--+-------------+-------------------------------------><
    |               |  |             |
    +- LISTLIBRARY -+  +-- library --+
    |               |
    +- LM ----------+
    |               |
    +- LISTMEMBERS -+
```

Description:

Use the LL (List Library) command to open a Library List window and optionally list the members of an MVS PDS/PDSE, VSE LIBR library or CMS minidisk.

The Library List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS.

Parameters:

library

The name of the library for which the contents are to be listed.

◊ For **MVS** the library parameter is a PDS (or PDSE) dataset name and optionally one or more member name mask.

A member name mask supports the following wild cards:

*	A single asterisk represents an entire member name or zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS(E) DSN and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by one or more blanks and/or a "," (comma). e.g.

```
LL  DEV.OEM.CBL202.CBLI.HELP.HTML(S*AN%  WIN*, *R)
```

◊ For **VSE** the library parameter can be:

1. A library name. In this case the statistics for the library are listed. e.g.

```
LL  CBLLIB
```

2. A library name and sublibrary name. In this case the sublibrary name may be a mask containing "*" (asterisk) wild cards as supported by VSE Librarian. The statistics for all sublibraries which fit the sublibrary name mask are listed. e.g.

```
LL  CBLLIB.TEST*
```

3. A library name, sublibrary name and member name and type. In this case the member name and type may be a mask containing "*" (asterisk) wild cards. The statistics for all members which fit the mask are listed. e.g.

```
LL  CBLLIB.TEST01.*.Z
```

This parameter is placed in the Library field of the Library List window.

Examples:

```
LL   CBL.JCL
```

List all members of the CBL.JCL PDS.

```
LL   PRD2.CBL.*.*
```

List all members of the PRD2.CBL LIBR sublibrary.

```
LL   PRD2.*
```

List all LIBR sublibraries of PRD2.

LLX

Syntax:

```
>>-- LLX -------------+-------------------+----------------------------------><
                      |                   |
                      +- | CLI Options | -+
```

CLI Options:

```
    +- -CMX -+   +------------------------------------------------------+
    |        |   v                                                      |
 >--+--------+---+--+-+-- lib_mask -+-+-+----------------------------+--+---->
    |        |      | |             | | |       +------------+       |
    +- -Q ---+      | +-- ddname ---+ | |       v            |       |
    |        |      |                 | +- ( -+-+- mbr_mask -+-+- ) -+
    +- -L ---+      |  (1)            |       |                |
                    +---- MACROPATH --+       +--- = ----------+
                    |                 |          (2)
                    |  (1)            |
                    +---- SYSAPF -----+
                    |                 |
                    |  (1)            |
                    +---- SYSLL ------+
                    |                 |
                    |  (2)            |
                    +---- = ----------+

 >--+---------------------------+--------------------------------------------->
    |                           |
    +-- SUBSET /where_clause/ --+
```

Notes:

1. Parameters MACROPATH, SYSAPF and/or SYSLL should be specified once only. If repeated the search will be repeated for all libraries in that concatenation.
2. "=" may be used if it is not the first library specification (*lib_mask*, *ddname*, etc.) or *mbr_mask* group referenced by the LLX command.

Description:

The LLX command invokes the Search for Library Members utility to locate members in PDS/PDSE libraries whose member names match the specified member mask(s), or, if executed with no parameters, opens the Search for Library Members Panel.

Output may be to a temporary CMX command file, which gets automatically displayed in a SELCOPY/i CBLe text edit window view, or multiple List Library Members windows, one for each library searched.

Parameters:

-CMX

Specifies that output format is an unsaved temporary file of DSN '%user%.LLX.Dyyyyddd.Thhmmss.TXT', with all information gathered in command file (CMX) format suitable for CMDTEXT point-and-shoot <PF4> edit of selected members.

This output format is default.

-Q

The -Q (Quiet) option implies -CMX, but suppresses pop-up windows that prompt the user for a decision to continue the search when either the number of members found in a library is greater than 999, or no members have been found in the last 10 libraries.

-L

The -L (List) option specifies that output format is multiple List Library Members windows. This option executes more quickly than -CMX (or -Q) but can result in a large number of open SELCOPY/i list window views.

lib_mask

Specifies a library DSN mask which identifies one or more PDS/PDSE libraries in which to search for the specified *mbr_mask* member mask(s).

lib_mask may optionally contain the following wild card characters:

`*`	A single asterisk represents an entire DSN qualifier or zero or more characters within a DSN qualifier mask.
`**`	A double asterisk represents zero or more qualifiers A double asterisk must be preceded by or follow a dot (period) and may not be used within a single DSN qualifier mask.
`%`	A single percent sign represents exactly one character within a DSN qualifier. (Up to 8 percent signs can be specified in each qualifier.)

If no *lib_mask* is specified, then at least one of *ddname*, MACROPATH, SYSAPF or SYSLL must be specified.

`ddname`

Specifies an existing MVS DDname which has been allocated to one or more PDS/PDSE library data set names. The library or library concatenation allocated to *ddname* will be searched for the specified *mbr_mask* member mask(s).

If no *ddname* is specified, then at least one of *lib_mask*, MACROPATH, SYSAPF or SYSLL must be specified.

`mbr_mask`

Specifies a member name mask which identifies one or more member names to be found. Multiple member masks may be entered on each specified *lib_mask*, *ddname* MACROPATH, SYSAPF and/or SYSLL parameter, constituting a member mask group.

mbr_mask may optionally contain the following wild card characters:

`*`	A single asterisk represents an entire member name or zero or more characters within a member name mask.
`%`	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS/PDSE *lib_mask* or *ddname* and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by one or more blanks and/or "," (commas).

The special character "=" (equals) may be used in place of a *mbr_mask* group in second and subsequent *mbr_mask* group specifications. This equates to be the *mbr_mask* group entered on the last *lib_mask*, *ddname*, MACROPATH, SYSAPF or SYSLL parameter.

Default is to locate **all** members in libraries identified by *lib_mask*, *ddname*, MACROPATH, SYSAPF or SYSLL.

`MACROPATH`

Indicates that all libraries in the user's current CBLe text editor macro path will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the CBLe text edit command, QUERY MACROPATH.)

`SYSAPF`

Indicates that all APF authorised load libraries will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the APF List Window (command SYSAPF.)

`SYSLL`

Indicates that all load libraries in the active Link List concatenation will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the Link List Window (command SYSLL.)

`=`

The special character "=" (equals) may represent a library specification (*lib_mask*, *ddname*, MACROPATH, SYSAPF or SYSLL) or a complete *mbr_mask* group.

"=" may be used in this capacity only if it is not the first library specification or *mbr_mask* group entered in the LLX command.

If used as a library specification, it is substituted with the previous library specification entered in the LLX command. If used as a *mbr_mask* group, it is substituted with the *mbr_mask* group belonging to the previous library specification entered in the LLX command.

`SUBSET /`*`where_clause`*`/`

The SUBSET parameter specifies a list window WHERE Clause (*where_clause*) used to apply additional search criteria on matching library member names.

The *where_clause* supports filter criteria only on field names returned by List Library Members windows for MVS load libraries and non-load libraries. See *"List Library Members"* for details of these field names, descriptions and their data types.

Examples:

`LLX   SYSEXEC(CBLII)  SYSPROC(=)  JGE.EXEC(=)`

Search all libraries in the SYSEXEC and SYSPROC concatenations, as well as library 'JGE.EXEC', for member name "CBLII".

`LLX -Q  CBL.JCL(SEL*)  LAC.JCL(*)`

Search library 'CBL.JCL' for all members beginning "SEL", and library 'LAC.JCL' for all members. The user will not be prompted to continue the search as a result of encountering > 999 member name matches in either library.

`LLX     CBL.**.JCL   SUBSET /WHERE LASTMOD => '2011/06/01'/`

Search all libraries with DSN matching the library DSN mask 'CBL.**.JCL' reporting all members that have been altered on or after 2011/06/01.

`LLX     STEPLIB(CBL*)   SUBSET /WHERE AC=1/`

Search all load libraries in the library concatenation allocated to DDname STEPLIB for module names beginning "CBL" that have been Link Edited with AC(1).

`LLX -L  NBJ.**.JCL`

Open a List Library Members window, one each for every library matching the DSN mask 'NBJ.**.JCL', which lists all members in that library.

LP

Syntax:

```
>>--+- LP ----------+---+--------+--+--------+---+--------------+------------><
    |               |   |        |  |        |   |              |
    +- LISTPATH ----+   +-- -C --+  +-- -S --+   +-- hfs_path --+
    |               |
    +- LISTP -------+
    |               |
    +- LPATH -------+
```

Description:

The LP (List Path entries) command may be used to open an HFS Path List window to list information about entries that match the specified HFS path.

If no parameters are specified, the list window will be opened with fields populated with parameters entered by the user for the last invocation of the HFS Path list window.

The HFS Path List window may also be opened via the LD command if the dataset specification begins with "." (dot/period) or contains "/" (slash), or via the List menu of the SELCOPY/i main window menu bar.

The LP command is not supported on CMS and VSE systems.

Parameters:

`-C`

Specify -C or -c to bypass case sensitivity for the **name** portion of the specified HFS path. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.

`-S`

Specify -S or -s to recursively list the contents of all sub-directories found within the HFS path specification.

hfs_path

Specifies the HFS path which is to be placed in the HFS Path> field of the list window. This may be a path name relative to the current working directory.

The following wild cards may only be specified within the name portion of the HFS path.

`*`	A single asterisk represents zero or more characters.
`%`	A single percent sign represents a single character.

Examples:

```
lp  -s     /u/johnd02/temp*
```

Lists the contents of the "/u/johnd02" directory where name begins with "temp" and, if a directory entry, list the contents of that directory and any of its sub-directories.

`listpath  '200401*_%%% Audit Report.tgz'`

List entries in the current working directory. The HFS path is quoted since the name mask contains a blank. Wildcards "*" and "%" are used to represent multiple (zero or more) and single characters respectively.

LQ

Syntax:

```
>>--+- LQ ------+--+--------------------------------+------------------------><
    |           |  |                                |
    +- LISTENQ -+  +- queuename -+------------------+
                                 |                  |
                                 +-- resourcename --+
```

Description:

Use the LQ (List MVS Enqueues) command to open an Enqueue List window and optionally list outstanding MVS enqueues by major name and minor name (queue name and resource name).

The Enqueue List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS or VSE.

Parameters:

queuename

The major name (queue name) of the ENQ resource. This is a 1-8 character upper case name. For example, dataset allocations are ENQueued with resource name SYSDSN.

This parameter is placed in the Queue Name field of the Enqueue List window.

resourcename

This is a 1-256 character, case sensitive minor name (resource name). You need only enter the prefix of the resources you are interested in. All resources for the given queue with resource beginning with this value are listed.

This parameter is placed in the Resource Name field of the Enqueue List window.

See Also:

LJQ Command

Examples:

`LQ   SYSDSN  SYS1`

List Enqueues for queue name SYSDSN for resource names starting with upper case 'SYS1'.

LV

Syntax:

```
>>--+- LV -------+--+-------------------------+----------------------------><
    |            |  |                         |
    +- LISTVTOC -+  +--- volume --+-----------+
                                  |           |
                                  +-- filter --+
```

Description:

Use the LV (List VTOC Files) command to open a VTOC File List window and optionally list, by data set name, entries contained in a DASD volume's Volume Table of Contents (VTOC).

The VTOC File List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS.

Parameters:

volume
The 1-6 character volume id containing the required VTOC.

This parameter is placed in the Volume field of the VTOC File List window.

filter
Note: The filter parameter is not yet implemented for VSE.

Select only catalog entries that match the specified filter mask. The filter mask supports the following wild cards:

`*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.

`**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.

`%` single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

A filter that does not contain an * (asterisk) wild card will be appended with *.** to list those data sets whose names begin with the filter string.

This parameter is placed in the Filter field of the VTOC File List window.

See Also:

LX Command
List VTOC entries by Extent.

Examples:

```
LV   CBLM02  CBL.SELC%%%.*
LV   CBLM02  SYS%.AX*.A*B*.**
```

LVOL

Syntax:

```
>>--+- LVOL ----+--+-------------+-------------------------------------------><
    |           |  |             |
    +- LISTVOL -+  +--- volume --+
```

Description:

Use the LVOL (List Volumes) command to open a DASD Volumes List window and optionally display the attributes of selected DASD volumes defined to your system.

The DASD Volumes window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Parameters:

volume
Select only defined DASD volumes that match the specified volume id mask. The volume mask supports the following wild cards:

`*` An asterisk indicates that one or more characters within the volume id can occupy that position. An asterisk can precede or follow a set of characters.

`%` A single percent sign indicates that exactly one character can occupy that position. (Up to 6 percent signs can be specified.)

A volume field that does not contain an * (asterisk) wild card will be appended with "*" to list all DASD volumes whose volume ids begin with the volume string.

This parameter is placed in the Volume field of the DASD Volumes window.

Examples:

```
LVOL *
```

List all volumes.

`LVOL SYS%%A`

List all volumes with 6 character volume name beginning with the characters 'SYS' and ending with 'A'.

LVR

Syntax:

```
>>-- LVR --+----------------------+-----------------------------------------><
           |                      |
           +--- cblvcat_syntax ---+
```

Description:

LVR opens the CBLVCAT Raw window and optionally executes CBLVCAT control statements.

The CBLVCAT Raw window may also be opened via the "Raw" menu item of the Execute CBLVCAT window.

Parameters:

cblvcat_syntax

Valid CBLVCAT syntax to be executed when the CBLVCAT Raw window is opened.
This parameter is placed in the "VCAT command line>" field of the CBLVCAT Raw window.

See Also:

VCAT Command

Examples:

```
LVR   listvcat key=nbj type=c

LVR   listvtoc vol=cblmct
```

LX

Syntax:

```
>>--+- LX ----------+--+------------+---------------------------------------><
    |               |  |            |
    +- LISTEXTENTS -+  +-- volume --+
```

Description:

Use the LX (List VTOC Extents) command to open a VTOC Extent List window and optionally list, by physical extent, the entries contained in a DASD volume's Volume Table of Contents (VTOC).

The VTOC Extent List contains an entry for each extent on the volume, including free extents and volume control areas such as the VTOC and the label area.

The VTOC Extent List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS or VSE.

Parameters:

volume

The 1-6 character volume id containing the required VTOC.

This parameter is placed in the Volume field of the VTOC Extent List window.

See Also:

LV Command

List VTOC entries by Data Set Name.

Examples:

```
LX    CBLM01
```

List extents on volume id CBLM01.

MAXIMISE

Syntax:

```
>>-+-- MAXIMISE --+--+--------------+---------------------------------------><
   |              |  |              |
   +- MAX --------+  +- windowname -+
```

Description:

This command maximises the specified window.

This command is equivalent to selecting the Maximise Button of the window to be maximised.

Parameters:

windowname

The window name of the window to maximise. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

MDINEXT

Syntax:

```
>>---- MDINEXT -------------------------------------------------------------><
```

Description:

For use in MDI applications only (e.g. CBLe and SELCOPY Debug), this command sets the focus window to be the next MDI child window in the ring of MDI child windows.

The MDI application's ring of child windows is maintained in creation sequence and wraps round from the last created to the first created.

MDIPREV

Syntax:

```
>>---- MDIPREV -------------------------------------------------------------><
```

Description:

For use in MDI applications only (e.g. CBLe and SELCOPY Debug), this command sets the focus window to be the previous MDI child window in the ring of MDI child windows.

The MDI application's ring of child windows is maintained in creation sequence and wraps round from the first created to the last created.

MINIMISE

Syntax:

```
>>-+-- MINIMISE --+--+--------------+----------------------------------------><
   |              |  |              |
   +- MIN --------+  +- windowname -+
```

Description:

This command minimises the specified window.

This command is equivalent to selecting the Minimise Button of the window to be minimised.

Parameters:

`windowname`
The window name of the window to minimise. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

MOVEWINDOW

Syntax:

```
                                               +- X=1 -+ +- Y=1 -+
                                               |       | |       |
                                        +- TO -+-------+-+-------+-+
                                        |      |       | |       | |
                                        |      +- X=n -+ +- Y=m -+ |
                                        |                          |
>>--+- MOVEWINDOW -+--+--------------+--+--------------------------+---------><
    |              |  |              |  |                          |
    +- MW ---------+  +- windowname -+  |      +- X=0 -+ +- Y=0 -+ |
                                        |      |       | |       | |
                                        +- BY -+-------+-+-------+-+
                                               |       | |       |
                                               +- X=n -+ +- Y=m -+
```

Description:

Use MOVEWINDOW to move the specified window to an absolute X,Y coordinate or to an X,Y coordinate relative to its current position. Coordinates 0,0 reference the top left corner of the desktop.

The current position of a window is defined to be the X,Y coordinate of the top left corner of the window.

Note that it is not possible to move a window completely outside the SELCOPY/i main window. At least one column of the title bar must be viewable within the desktop. Therefore, limits governed by the currnet window position and the 3270 session screen size are imposed on the X,Y values supplied on the MOVEWINDOW command. Specifying X,Y values that fall outside these limits result in the limit value being used.

A window may also be moved by dragging then dropping the window using the titlebar. Similarly, the default function key table for the TitleBar level contains MOVEWINDOW commands allocated to PFkeys PF07, PF08, PF10 and PF11 to position the window up and down 1 line and left and right 1 column respectively.

Parameters:

`windowname`
The window name of the window to be moved. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

`TO`
Indicate that an absolute X,Y position follows.

`BY`
Indicate that a relative X,Y position follows.

`X=n`
Define the horizontal coordinate.

If absolute, *n* must be a positive integer. The window will be moved horizontally so that the top left corner of the window is in column *n*.
Default value for *n* is 1.

If relative, *n* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative horizontal displacement. The window will be moved horizontally *n* columns to the right (positive) or left (negative) from its current position.
Default value for *n* is 0.

`Y=`*m*

Define the vertical coordinate.

If absolute, *m* must be a positive integer. The window will be moved vertically so that the top left corner of the window is in row n.
Default value for *m* is 1.

If relative, *m* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative vertical displacement. The window will be moved vertically *m* rows downwards (positive) or upwards (negative) from its current position.
Default value for *m* is 0.

Examples:

`MOVEWINDOW  TO X=32`

Window is moved to column 32, row 1 (Y=1 is default).

`MW TO X=32 Y=80`

Window is moved to column 32, row 80. However, if row 80 is outside the 3270 display, then the window is moved so that the title bar is displayed in the last visible row.

`MW BY X=2 Y=-5`

Window is moved 2 columns to the right and 5 columns upwards.

See Also:

SIZEWINDOW

NEXTMAINWINDOW

Syntax:

```
>>-+- NEXTMAINWINDOW -+-----------------------------------------------------><
   |                  |
   +- NMW ------------+
```

Description:

This command sets the focus window to the next main window i.e. one that is an immediate child of the desktop window. e.g. instances of CBLe, SELCOPY Debug, CBLVCAT Interactive or any list windows/dialogs created directly from the desktop menu bar or command line.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

See Also:

PREVMAINWINDOW
NEXTWINDOW
PREVWINDOW
MDINEXT
MDIPREV

NEXTWINDOW

Syntax:

```
>>--+- NEXTWINDOW -+--------------------------------------------------------><
    |              |
    +- NW ---------+
```

Description:

This command sets the focus window to the next window in the ring of all windows.

The ring is maintained in creation sequence and wraps round from the last created to the first created.

POWER

Syntax:

```
>>-- POWer ---+-----------------+-------------------------------------------><
              |                 |
              +- power_command -+
```

Description:

For VSE only, use the POWER command to open a POWER Command Output window and optionally execute a VSE POWER command.

The Power Command Output window may also be opened via the File menu of the SELCOPY/i main window menu bar.

If SELCOPY/i INI variables System.VSESMLogon=No (i.e. no Security Manager is active) and System.TrustedUser=No, then POWER commands are restricted to PDISPLAY operations only.

Parameters:

power_command

Any supported VSE POWER command.

This parameter is placed in the POWER Command field of the POWER Command Output window.

Note that some POWER commands are not supported for cross partition usage (e.g. PDISPLAY STATUS)

Examples:

`POWER D LST`

Display the POWER list queue.

`POW  PRELEASE RDR,CBLTEST`

Release entry CBLTEST from the POWER reader queue.

PREVMAINWINDOW

Syntax:

```
>>--+- PREVMAINWINDOW -+----------------------------------------------------><
    |                  |
    +- PMW ------------+
```

Description:

This command sets the focus window to the previous main window i.e. one that is an immediate child of the desktop window, e.g. instances of CBLe, SELCOPY Debug, CBLVCAT Interactive or any list windows/dialogs created directly from the desktop menu bar or command line.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

Because MDI applications such as the CBLe editor and SELCOPY Debug have many child windows of their own (navigable with MDINext/Prev commands), this command is necessary to switch directly between SELCOPY/i application windows.

By default, PF09 is set to **MDINext**.
By default, PF21 is set to **PrevMainWindow** (Shift-PF9)

See Also:

NEXTMAINWINDOW
NEXTWINDOW
PREVWINDOW
MDIPREV

MDIPREV

PREVWINDOW

Syntax:

```
>>--+- PREVWINDOW -+---------------------------------------------------------><
    |              |
    +- PW ---------+
```

Description:

This command sets the focus window to the previous window in the ring of all windows.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

QUIT

Syntax:

```
>>-- QUIT -------------------------------------------------------------------><
```

Description:

Use QUIT to exit and close the current SELCOPY/i window.

If the current window is the SELCOPY/i main window, then a pop-up window prompts the user to confirm whether or not to quit the SELCOPY/i session.

RENAME

Syntax:

Rename an MVS data set, HFS file or PDS(E) member, or a CMS file on an accessed minidisk:

```
>>-- REName ------------------------ fileid1 --------- fileid2 --------------><
```

Rename a VSE sequential or VSAM file:

```
>>-- REName ----+- volid ---+-- : -- fileid1 --------- fileid2 --------------><
                |           |
                +- catdsn --+
```

Description:

Rename a single sequential DASD file, HFS file, PDS(E) member or VSAM file.

To succeed, the user must have sufficient read/write authority for the file and no exclusive ENQ or LOCK should already exist for the file.

In an MVS environment, when renaming a PDS member, parameters should be specified in one of the following formats:

1. fileid1 is the data set and member name of the member to be renamed and fileid2 is the new member name only.
2. fileid1 is the **quoted** data set and member name for the member to be renamed and fileid2 is the **quoted** data set and new member name.

This second method also applies to MVS sequential and VSAM data sets whereupon RENAME executes the following IDCAMS command:

```
   ALTER  fileid1  NEWNAME(fileid2)
```

Therefore, types of file that may be renamed and the supported format of fileid1/2 is governed by the IDCAMS ALTER command. (See "DFSMS Access Method Services for Catalogs".)

For HFS files, specification of a leading "." (dot/period) or "/" (slash) in the HFS path name is mandatory in order to distinguish it from an MVS data set name. Both fileid1 and fileid2 may be specified as an absolute or relative HFS path and may reference a file name, directory name, hard link or symbolic link.

For VSE, sequential files may only be renamed if the CBL software product **CBLVCAT** is licensed. SELCOPY/i uses CBLVCAT's MOD operation to perform the rename.

Parameters:

volid
: For VSE sequential disk files, this is the volume serial number of the DASD volume on which the sequential file resides.

catdsn
: For VSE VSAM files, this is the full fileid of the VSAM catalog to which the VSAM managed file belongs.

fileid1
: The current fileid in full of the file to be renamed. For HFS, this may be an absolute or relative path name.

fileid2
: The new fileid to be assigned to the file. For HFS, this may be an absolute or relative path name.

Examples:

```
rename  cbl.ssc.ctl(sstest)    sstest01
```

Rename an MVS PDS(E) member.

```
rename "cbl.jcl(cblins01)"    "cbl.jcl(install)"
```

Rename an MVS PDS(E) member.

```
ren     cbl.cbli.test.file     nbj.test.data
```

Rename an MVS sequentail or VSAM data set.

```
rename   SYSWK1:CBL.SELCOPY.NAM    CBL.SELCOPY.NAM.NEWNAME
```

Rename a VSE sequential disk file. (CBLVCAT must be licensed.)

```
rename   VSESP.USER.CATALOG:CBL.TEST.KSDS   CBL.TEST.KSDS.NEWNAME
```

Rename a VSE VSAM managed data set.

```
rename   ./nbj.tmp.gz    /scr/install.x1832.gzip
```

Rename an HFS file and move it to a new directory.

RESTORE

Syntax:

```
>>-+-- RESTORE --+--+--------------+----------------------------------------><
   |             |  |              |
   +- RES -------+  +- windowname -+
```

Description:

This command restores the specified window from a maximised or minimised state back to its original size and position.

This command is equivalent to selecting the Restore Button of the window to be minimised.

Parameters:

windowname
: The window name of the window to maximise. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

RETRIEVE

Syntax:

```
>>--- RETRIEVE ----+--  + (plus) ---+---------------------------------------><
                   |                |
                   +--  - (minus) --+
```

Description:

For each window, SELCOPY/i stores a history of the commands issued. Use the RETRIEVE command to recall commands from the ring of executed commands.

The recalled command is placed on the focus window's command line.

By default this command is assigned to function key **PF12**.

Parameters:

+

Recall commands scrolling forwards through the ring.

-

Recall commands scrolling backwards through the ring.

RIGHT

Syntax:

```
>>-- RIGHT -+--------------+--+-----------+---------------------------------><
            |              |  |           |
            +- windowname -+  +- CURSOR --+
                              |           |
                              +- DATA ----+
                              |           |
                              +- HALF ----+
                              |           |
                              +- MAX -----+
                              |           |
                              +- PAGE ----+
                              |           |
                              +- n_cols --+
```

Description:

Scroll the view of the data within the specified window right towards the last column of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_cols* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, RIGHT, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to RIGHT is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one column is used.

List windows may contain fields that have **KEY** attribute **YES** defined in the Field Descriptor Block. Fields with this attribute are always in view and may not be scrolled right or left. If the cursor is positioned in a column belonging to this type of field, then, for LEFT CURSOR and RIGHT CURSOR, the cursor is considered to be outside the display area. All columns of data that do not belong to a **KEY** field are scrollable using LEFT and RIGHT.

By default this command is assigned to function key **PF11**.

Parameters:

windowname

The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

`CURSOR`
: The scrollable column on which the cursor is positioned becomes the first scrollable column of the display.
If the cursor is positioned outside the display area, in a KEY field or on the first scrollable column within the display area, then RIGHT PAGE is executed instead.

`DATA`
: Scroll right so that the last scrollable column in the current display area becomes the first scrollable column of the display.

`HALF`
: Scroll a number of columns so that the column situated half way along the width of the current display of scrollable columns, becomes the first scrollable column of the display.

`MAX`
: Scroll right to display the last scrollable column of data.
Where the display area is able to contain all columns of data, the first scrollable column becomes the first scrollable column of the display. Otherwise, the last scrollable column of data becomes the last column of the scrolled display.

`PAGE`
: Scroll right so that the column of data to the right of the last scrollable column in the current display, becomes the first scrollable column of the display.

`n_cols`
: Scroll right a specified number of columns.
The column of data that is *n_cols* to the right of the first scrollable column becomes the new first scrollable column of the display.

SDATA

Syntax:

```
>>-- SData -- sde_command ---------------------------------------------------><
```

Description:

Direct a command to the SELCOPY/i Structured Data Environment (SDE).

The SDATA command allows SDE commands to be issued from any SELCOPY/i window.
If the CBLe text editor main window has been stopped, SDATA will start the CBLe main window and open an edit view for the user's HOME CMX file before executing the SDATA command.
Also see the SDATA CBLe CLI command.

Parameters:

`sde_command`
: Any SDE command.

SDE

Syntax:

```
>>-- SDE -----+-----------------+------------------------------------------><
              |                 |
              +-- sde_command --+
```

Description:

Performs the same action as SDATA except that, if no arguments are specified the Structured Edit dialog window is opened.

Parameters:

`sde_command`
: Any SDE command.
Default is EDITDIALOG.

SDSF

Syntax:

```
>>--- SDSF ------------------------------------------------------------------><
```

Description:

When running SELCOPY/i in TSO (with or without ISPF), the SELCOPY/i command **SDSF** starts the System Display and Search Facility.

SDSF is started as a full screen TSO application which returns control to SELCOPY/i only after it is closed.

SDSF may also be started via the **Utilities** menu of the SELCOPY/i main window menu bar.

SELCOPY

Syntax:

```
>>-- SELCopy ---------------------------------------------------------------><



>>-- SELCopy --- -CTL selcctl --+------------------+--+------------------+-->
                                |                  |  |                  |
                                +-- -LIB libpath --+  +-- -PGM loadmod --+

 >--+-------------------------------------------------------------------+---><
    |                                                                   |
    |               +-- -DLI --+                                        |
    |               |          |                                        |
    +-- -PSB psbn --+----------+--+----------------+--+---------------+-+
                    |          |  |                |  |               |
                    +-- -BMP --+  +-- -IMSID ssn --+  +-- -AGN grpn --+
```

Description:

Use the SELCOPY command to open the SELCOPY Debug window and load the specified file containing SELCOPY control statements.

The SELCOPY Debug window may also be opened by selecting 'SELCOPY Debug/Dev' from the File menu in the CBLe main window menu bar.

If no parameters are specified, then a pop-up window prompts the user enter a SELCOPY control statement source fileid.

SELCOPY Debug opens a CBLe edit view for the **selcctl** SYSIN/SYSIPT source file input. If selcctl is supplied as an explicit fileid (DSN) which is not locked (with an exclusive ENQ) by another process, and the user has sufficient authority, the file is edited read-write. In this case, updates can be made to the SYSIN source and execution repeated without leaving the SELCOPY Debug application.

Where **selcctl** refers to a previously allocated filename (DD/FILEDEF/DLBL), which may be a concatenation of datasets, then it is opened read-only by the SELCOPY Debug SYSIN/SYSIPT CBLe edit view. A token of 8 or fewer characters with no embedded dots is automatically treated an allocated filename.

Read-only access to SELCOPY control cards held in CA-LIBRARIAN members is made available on MVS systems via SELCOPY/i's ALLOC, which supports the SUBSYS(LAM) parameter.

See SELCOPY Debug & Development for full information on the features of the SELCOPY Debug application, including point-and-shoot options provided through the PF4 key (SdbPopup).

MVS JCL decks

SELCOPY Debug handling of MVS JCL decks can be acheived using the JCLCMX pre-processor tool. JCLCMX is a REXX macro that must be be run from within the CBLe file editor.

Simply edit the JCL deck using CBLe, then key in JCLCMX from the edit command line. The tool will generate a SELCOPY/i ALLOC command corresponding to each DD statement, and a CALL command corresponding to each EXEC statement. Where

EXEC PGM=SELCOPY is encountered, the appropriate SELCOPY Debug invocation command is generated.

The commands generated by JCLCMX are not immediately executed, but presented to the user in separate, editable CMX files, one for each job step, ready for individual point-and-shoot execution using the PF4 key (CMDTEXT) in the standard SELCOPY/i fashion.

Parameters:

`-CTL` *`selcctl`*
: The full fileid or an already defined DD/FILEDEF/DLBL of the file containing the SELCOPY control statements.

`-LIB` *`libpath`*
: For **MVS** SELCOPY jobs only, a list of load libraries to be included before the current environment's search library chain. This is equivalent to supplying a JCL STEPLIB statement in a batch job and so may be used to control which SELCOPY module is executed and also any routines executed via the SELCOPY CALL statement.

Libpath may be one of the following:

- ◊ A DDname which has been pre-allocated to one or more load libraries.
- ◊ One or more load library DSNs separated by '`,`' (commas), '`;`' (semi-colons) or '` `' (blanks).
  Note that if blanks are used, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

`-PGM` *`loadmod`*
: Use an alternative load MODULE or PHASE name in place of the default name, SELCOPY.

`-PSB` *`psbn`*
: For **MVS IMS** SELCOPY jobs only, the Program Specification Block name to be used to access DL1 data.

`-DLI`
`-BMP`
: For **MVS IMS** SELCOPY jobs only, use either the DLI region or BMP region.
Default is -DLI.

`-IMSID` *`ssn`*
: For **MVS IMS** SELCOPY jobs only, the sub-system name of the IMS Region to which to connect.

`-AGN` *`grpn`*
: For **MVS IMS** SELCOPY jobs only, the IMS Application Group name.

Examples:

```
SELC     -CTL CBL.SSC.CTL(DIRD01)
```
Start the SELCOPY Debug window and load control statements from the DSN CBL.SSC.CTL(DIRD01).

```
SELCOPY  -CTL SYSIN
```
Start SELCOPY Debug using control statements in the currently allocated filename SYSIN.

```
SELCOPY  -CTL SYS3.CBL.SELCOPY.CTL001   -LIB  "DEV.CBL.LOAD   DEV.TEST.RTN001.LOAD"
```
Start SELCOPY Debug using control statements in data set SYS3.CBL.SELCOPY.CTL001. Search load libraries DEV.CBL.LOAD then DEV.TEST.RTN001.LOAD ahead of the search chain when locating the SELCOPY executable module and any modules called using the SELCOPY CALL operation.

```
<alloc  dd(INCTL)  dsn('SYS3.NBJ.EQU001'  'SYS3.TEST.SELCOPY.CTL(SQ10249)')  shr
<selcopy  -ctl INCTL
```
In this example, the CBLe ALLOCATE command is first used to allocate a concatenation of two DSNs to DDname, INCTL. These commands should be entered in a CBLe edited CMX file and executed using CMDTEXT (PF4).

SETCOLOUR

Syntax:

```
                        +-------------------+
                        v                   |
>>--+- SETCOLOUR -+--+- ttref -- ttval -+------------------------------------><
    |             |
    +- SC --------+
```

Description:

Use SETCOLOUR to remap the appearence of a colour and its associated highlighting.

SELCOPY/i maintains a translate table that defines how a colour/highlight style combination is to be displayed. Each colour/highlight style combination may be remapped so that it is displayed as a different colour/highlight style.

A colour/highlight style combination is represented as a pair of descriptor characters, the first of which is the initial of the colour, and the second the initial of a highlight style. All valid combinations are as follow:

		2nd Char			
		B	**D**	**R**	**U**
1st Char	**B**	Blue Blink	Blue Default	Blue REVVideo	Blue Underline
	G	Green Blink	Green Default	Green REVVideo	Green Underline
	P	Pink Blink	Pink Default	Pink REVVideo	Pink Underline
	R	Red Blink	Red Default	Red REVVideo	Red Underline
	T	Turquoise Blink	Turquoise Default	Turquoise REVVideo	Turquoise Underline
	W	White Blink	White Default	White REVVideo	White Underline
	Y	Yellow Blink	Yellow Default	Yellow REVVideo	Yellow Underline

These descriptor pairs reference cells in the translate table as well as describing the values within a cell. The default translate table is such that each cell referenced by a colour/highlight combination contains that colour/highlight combination. i.e.

		2nd Char			
		B	**D**	**R**	**U**
1st Char	**B**	BB	BD	BR	BU
	G	GB	GD	GR	GU
	P	PB	PD	PR	PU
	R	RB	RD	RR	RU
	T	TB	TD	TR	TU
	W	WB	WD	WR	WU
	Y	YB	YD	YR	YU

A number of **ttref** and **ttval** pairs may be specified, each defining a single update to the translate table.

Parameters:

ttref

A colour/highlight style pair referencing a cell in the translate table.

ttval

A colour/highlight style pair to be inserted into the translate table cell referenced by *ttref*. Where SELCOPY/i uses the *ttref* colour/highlight style combination, the *ttval* colour/highlight style is displayed instead.

Examples:

```
SETCOLOUR   BD BR
```

Blue Default will be displayed as Blue REVVideo.

```
SC     RR YB     PD TU
```

Red REVVideo will be displayed as Yellow Blink and Pink Default will be displayed as Turquoise Underline.

SETFOCUS

Syntax:

```
>>--+- SETFOCUS -+---+--------------+--------------------------------------><
    |            |   |              |
    +- SF -------+   +- windowname -+
```

Description:

Use the SETFOCUS command to change the focus window.

Parameters:

windowname

The name of the window to receive the focus. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

SHOWPOPUPMENU

Syntax:

```
>>--+- SHOWPOPUPMENU -+---+-------------+------------------------------------><
    |                 |   |             |
    +- SPM -----------+   +-- winname --+
```

Description:

The SHOWPOPUPMENU command displays the options popup menu for the current storage display window.

Storage display windows include SELCOPY Debug Work Area and POS windows, CBLe Hex display windows and the CBLNAME window.

By default, the SHOWPOPUPMENU command is assigned to PF5 in storage display windows. The options popup menu may also be opened via the system menu button of the storage display window.

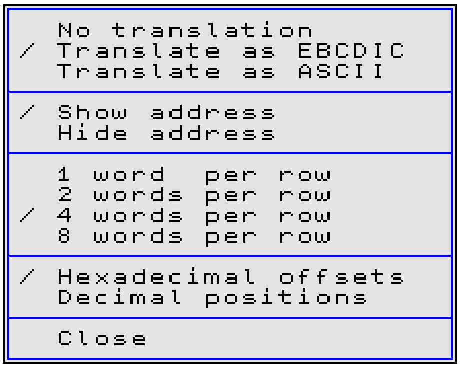


Figure 132. Storage Window Popup.

The mark "/" against items in the menu identifies the current status of the storage display window.

`No translation`
`Translate as EBCDIC`
`Translate as ASCII`
: Defines the interpretation of the hexadecimal data in the character field. (i.e. ASCII or EBCDIC.) If No translation is selected, then the character field is suppressed.

`Show address`
`Hide address`
: Defines whether the field containing the address in storage of each row of data is displayed or suppressed.

`1/2/4/8 words per row`
: Defines the number of words (length 4 bytes) are displayed in each row of data.

`Hexadecimal offsets`
`Decimal Positions`
: Defines whether the numeric field, displaying the displacement of each row of data relative to the first byte of data in the storage window, is presented as a hexadecimal offset or as a decimal position.
(e.g. row displayed as hexadecimal offset X'0000f0' is equivalent to decimal position 241.)

`Close`
: Closes the storage display window.

Parameters:

`winname`
: The name of the storage window for which the popup menu will apply. If not supplied then the name of the window in which the command is issued (via a command line or a function key) is assumed.

SHOWWATTR

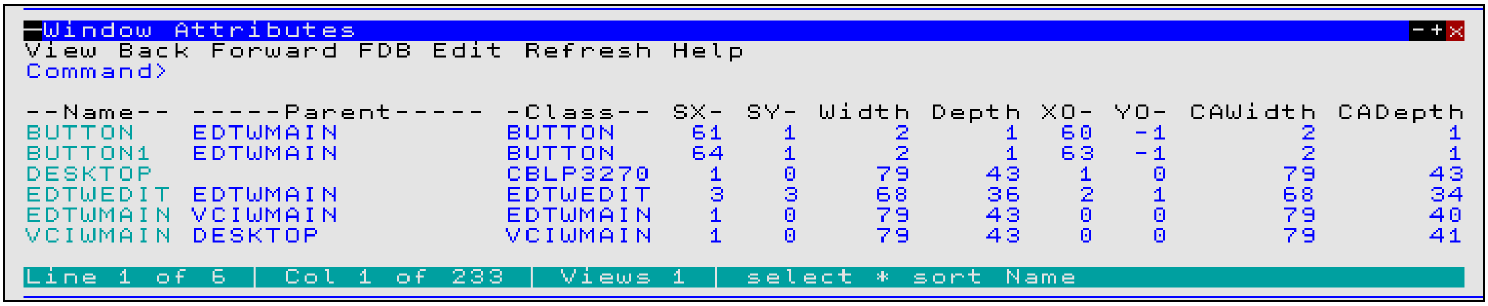
Syntax:

```
>>--+- SHOWWATTR -+-----------------------------------------------------------><
    |             |
    +- SWA -------+
```

Description:

Use the SHOWWATTR command to open the **Window Attributes** window to display the attributes of all open windows.

The Window Attributes window is essentially a List window and has the same charactersitics as List windows. For example select, sort and filter to display new views of the data are supported.



```
Window Attributes                                                                 -+x
View Back Forward FDB Edit Refresh Help
Command>

--Name-- -----Parent----- -Class-- SX- SY- Width Depth XO- YO- CAWidth CADepth
BUTTON   EDTWMAIN         BUTTON    61   1     2     1  60  -1       2       1
BUTTON1  EDTWMAIN         BUTTON    64   1     2     1  63  -1       2       1
DESKTOP                   CBLP3270   1   0    79    43   1   0      79      43
EDTWEDIT EDTWMAIN         EDTWEDIT   3   3    68    36   2   1      68      34
EDTWMAIN VCIWMAIN         EDTWMAIN   1   0    79    43   0   0      79      40
VCIWMAIN DESKTOP          VCIWMAIN   1   0    79    43   0   0      79      41

Line 1 of 6 | Col 1 of 233 | Views 1 | select * sort Name
```

Figure 133. Window Attributes Window.

Columns Displayed:

Name	Type	Description
Name	Char	Window name
Parent	Char	Window parent name
Class	Char	Window class
SX	Int	Screen x coordinate
SY	Int	Screen y coordinate
Width	Int	Window width
Depth	Int	Window depth
XO	Int	X offset within parent
YO	Int	Y offset within parent
CAWidth	Int	Client area width
CADepth	Int	Client area depth
CAXO	Int	Client area x offset
CAYO	Int	Client area y offset
CursorX	Int	Cursor x coordinate
CursorY	Int	Cursor y coordinate
Title	Char	Window title

SIZEWINDOW

Syntax:

```
                                                +- TO -+
                                                |      |   (1)
>>-+- SIZEWINDOW -+--+---------------+--+------+--+- D=n -- W=m -+---><
   |              |  |               |  |      |  |              |
   +- SZW --------+  +- windowname -+  +- BY -+  +- D=n --------+
                                                  |              |
                                                  +-------- W=m -+
```

Notes:

1. D= and W= parameters may be entered in any order.

Description:

Use SIZEWINDOW to resize the specified window to an absolute depth and/or width or to a depth and/or width relative to the window's current size.

The window depth and window width are the number of rows and columns respectively that are displayed in the window's displayable area.

Note that the window's displayable area includes items such as the title bar and command line, but do not include the window's horizontal and vertical borders which occupy an additional 2 rows and 4 columns respectively. e.g. To determine the number of rows/columns of text that would be displayed on resizing a CBLe edit view, deduct 2 from the depth value (title bar, command line) and, if SCALE is set on, deduct an extra 1 for the scale line. If PREFIX is set on, deduct 6 columns from the width value for the prefix area.

The current position of the window is unchanged following a SIZEWINDOW command.

A window may also be resized by dragging the border of the window.

Parameters:

`windowname`
The window name of the window to be resized. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

`TO`
Indicate that an **absolute** depth and/or width follows. (Default)

`BY`
Indicate that a **relative** depth and/or width follows.

`D=n`
Define the window depth change.

If absolute, *n* must be a positive integer. The number of rows displayed is equal to n and the bottom border of the window is raised or lowered accordingly.

If relative, *n* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative adjustment to the number of rows displayed. The number of rows displayed is increased or decreased by *n* rows and the bottom border of the window is raised or lowered accordingly.

There is no default value for *n*. If omitted, the depth is unchanged.

`W=m`
Define the window width change.

If absolute, *m* must be a positive integer. The number of columns displayed is equal to m and the right border of the window is adjusted accordingly.

If relative, *m* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative adjustment to the number of columns displayed. The number of columns displayed is increased or decreased by *m* columns and the right border of the window is adjusted accordingly.

There is no default value for *m*. If omitted, the width is unchanged.

Examples:

```
SIZEWINDOW  TO D=21
```
Focus window is resized to display 21 rows. Number of columns is unaltered.

```
SZW W=33 D=14
```
Focus window is resized to display 14 rows and 33 columns.

```
SZW BY D=8 W=-4
```
Focus window is resized to display an additional 8 rows and 4 columns less.

See Also:

MOVEWINDOW
SET WINSIZE CBLe command for resizing the current CBLe edit view or frame window.

SQL

Syntax:

```
>>-- SQL --+-------------------+--+-------------------+--+-----------------+->
           |                   |  |                   |  |                 |
           +- -SSN=ssn_name ---+  +- -PLAN=plan_name -+  +- -LIMit=n_rows -+

           +- -EXec=Immediate -+  +- -COmmit=Yes -----+
           |                   |  |                   |
 >-----------------------------+-------------------------+-----------------+-><
           |                   |  |                   |  |                 |
           +- -EXec=Delay -----+  +- -COmmit=No ------+  +- sql_syntax ----+
                                                         |                 |
                                                         +- < - sql_ctl ---+
```

Description:

Use the SQL command to open the Dynamic SQL window.

The Dynamic SQL window may also be opened via the File menu of the SELCOPY/i main window menu bar.

The Dynamic SQL window connects the user to the DB2 subsystem using the DB2 subsystem name and plan specified in the **CBLNAME** load module.

Note: Not implemented for CMS or VSE.

Parameters:

`-SSN=`*`ssn_name`*
The DB2 sub-system to be the target of the CONNECT.
This parameter is placed in the "DB2 Subsystem>" field of the Dynamic SQL window.
Default is that defined by the SELCOPY/i INI option, DB2.SSN, otherwise the sub-system name specified in the DB2SubSys field of the CBLNAME load module is used.

`-PLAN=`*`plan_name`*
The SELCOPY DB2 plan name which has been bound to the DB2 sub-system.
This parameter is placed in the "Plan>" field of the Dynamic SQL window.
Default is that defined by the SELCOPY/i INI option, DB2.Plan, otherwise the plan name specified in the DB2Plan field of the CBLNAME load module is used.

`-LIMIT=`*`n_rows`*
Limit the number of rows to be displayed in the Dynamic SQL window following a SELECT transaction. Once the limit threshold has been reached, a pop-up message window is displayed and no further attempt is made to retrieve selected rows of data.
The *n_rows* value is placed in the "Select Limit>" field of the Dynamic SQL window.
The default limit is that defined by the SELCOPY/i INI option, DB2.SelectLimit, otherwise no limit is implied.

`-EXEC=IMMEDIATE|DELAY`
Determine whether the SQL command is to be executed immediately when the Dynamic SQL window is opened or simply placed on the SQL Statement command line.
The default is IMMEDIATE.

`-COMMIT=YES|NO`
Determine whether a COMMIT is to be automatically issued following every transaction (AutoCommit). If COMMIT=NO, then the user should issue COMMIT manually to commit any changes made to the data. A commit is executed automatically when the Dynamic SQL window is closed, regardless of the AutoCommit field setting.
The commit value is reflected in the "AutoCommit>" field of the Dynamic SQL window.
The default is YES.

`sql_syntax`
Valid SQL syntax to be executed when the Dynamic SQL widow is opened.
The *sql_syntax* string is placed in the first SQL Statement line field of the Dynamic SQL window.

`<` *`sql_ctl`*
Input contol file containing one or more valid SQL statements to be executed when the Dynamic SQL widow is opened.
The "< *sql_ctl*" string is placed in the first SQL Statement line field of the Dynamic SQL window.

Examples:

```
SQL -SSN=DB8G  select * from dsn810.emp
```
Display all entries in the table DSN810.EMP of DB2 subsystem DB8G.

```
SQL -SSN=CBLA  -LIMIT=200  < CBL.SQL.CTL(TAB0326)
```
Execute in subsystem CBLA, all SQL statements provided via library mamber CBL.SQL.CTL(TAB0326) and limit the number list rows displayed to 200.

SVC

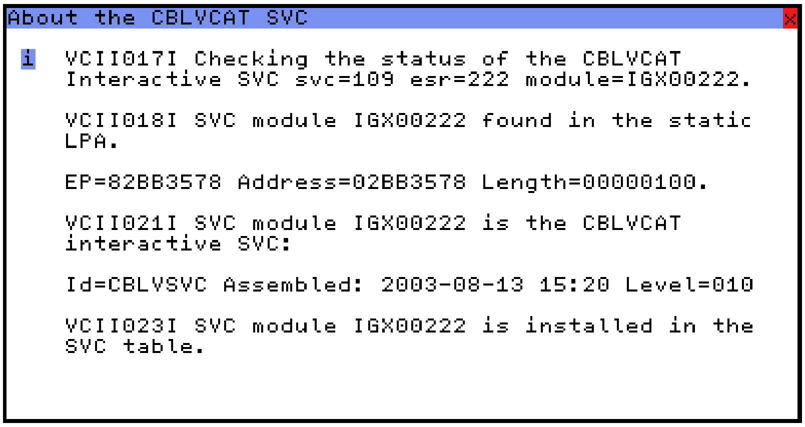
Syntax:

```
>>--- SVC -------------------------------------------------------------------><
```

Description:

For MVS only, use the SVC command to display the CBLVCAT SVC window containing the current status of the CBLVCAT Interactive (VCI) SVC required for LISTVCAT operations.

The CBLVCAT SVC window may also be opened via the System menu of the SELCOPY/i main window menu bar.



```
VCII017I Checking the status of the CBLVCAT
Interactive SVC svc=109 esr=222 module=IGX00222.

VCII018I SVC module IGX00222 found in the static
LPA.

EP=82BB3578 Address=02BB3578 Length=00000100.

VCII021I SVC module IGX00222 is the CBLVCAT
interactive SVC:

Id=CBLVSVC Assembled: 2003-08-13 15:20 Level=010

VCII023I SVC module IGX00222 is installed in the
SVC table.
```

Figure 134. CBLVCAT SVC Window.

SYSAPF

Syntax:

```
>>-- SYSAPF -----------------------------------------------------------------><
```

Description:

Use the SYSAPF command to open the APF List window.

The APF List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not valid for CMS and VSE.

SYSCOMMAND

Syntax:

```
>>--+- SYSCOMMAND -+--- command ---------------------------------------------><
    |              |
    +- SYS --------+
    |              |
    +- SYSTEM -----+
    |              |
    +- TSO --------+
    |              |
    +- CMS --------+
    |              |
    +- DOS --------+
```

Description:

Pass the command directly to the local CMS or TSO environment for execution.

When a command is issued in a SELCOPY/i window, the following occurs:

1. If the command is recognised as a SELCOPY/i command it is executed by SELCOPY/i.
2. If the command is not recognised as a SELCOPY/i command, it is passed to the CMS or TSO environment.

Parameters:

`command`
: Valid CMS or TSO command or expression.

Example:

`cms query dasd`
: Pass the command "query dasd" to CMS.

SYSI

Syntax:

```
>>--+- SYSI -----+-----------------------------------------------------------><
                 |            |
                 +- SYSINFO -+
```

Description:

Use the SYSI command to open the Operating System window.

The System Information window may also be opened via the System menu of the SELCOPY/i main window menu bar.

SYSLL

Syntax:

```
>>-- SYSLL ------------------------------------------------------------------><
```

Description:

Use the SYSLL command to open the Link List window.

The Link List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not valid for CMS and VSE.

SYSLPA

Syntax:

```
>>-- SYSLPA -----------------------------------------------------------------><
```

Description:

Use the SYSLPA command to open the LPA Modules window.

The LPA Modules window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not valid for CMS and VSE.

SYSMENU

Syntax:

```
>>-- SYSMENU --+--------------+----------------------------------------------><
               |              |
               +- windowname -+
```

Description:

Use the SYSMENU command to open the System Menu for the specified window.

The System Menu may also be opened via the System Menu button of a window.

Parameters:

windowname
The window name of the window for which the system menu is to be opened. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

SYSPGM

Syntax:

```
>>-- SYSPGM -----------------------------------------------------------------><
```

Description:

Use the SYSPGM command to open the Loaded Programs window.

The Loaded Programs window may also be opened via the System menu item of the SELCOPY/i main window menu bar.

SYSSTOR

Syntax:

```
>>-- SYSSTOR ----------------------------------------------------------------><
```

Description:

Use the SYSSTOR command to open the Storage Statistics window.

The Storage Statistics window may also be opened via the System menu of the SELCOPY/i main window menu bar.

SYSTASK

Syntax:

```
>>-- SYSTASK ----------------------------------------------------------------><
```

Description:

Use the SYSTASK command to open the Task List window.

The Task List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS and VSE.

TASK

Syntax:

```
>>-- TASK --- pgmname ---+------------------+---+----------------+---><
                         |                  |   |                |
                         +-- -LIB libpath --+   +-- -PARM parm --+
```

Description:

For MVS only, use the TASK command to start a program as a sub-task of SELCOPY/i.

TASK commands are generated by the CBLe REXX macro, JCLCMX, to run non-SELCOPY job steps of an MVS batch job in the environment in which SELCOPY/i is being executed (i.e. TSO or VTAM).

Parameters:

pgmname
The name of the program load module to be executed.

-LIB *libpath*
A list of load libraries to be included before the current environment's search library chain. This is equivalent to supplying a JCL STEPLIB statement in a batch job and so may be used to define the location of the program module to be executed plus any modules called by the program.

Libpath may be one of the following:

◊ A DDname which has been pre-allocated to one or more load libraries.
◊ One or more load library DSNs separated by ',' (commas), ';' (semi-colons) or ' ' (blanks).
Note that if blanks are used, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

-PARM *parm*
Parameter string to be passed to the program. This is equivalent to supplying the PARM parameter on an JCL EXEC statement in a batch job.

If the parm string contains blanks, then quotes must be used to delimit the parm string.

Examples:

```
TASK  TRSMAIN   PARM='UNPACK'
```

Start program TRSMAIN to unpack a tersed data set.
Relevant INFILE and OUTFILE ddnames must be allocated before executing this command. (See the CBLe command ALLOCATE.)

```
TASK  MYPROG  -LIB "SYS7.DEV.MYLIB.LOAD  SYS4.USER.ROUTINES.X01323"
```

Include the specified libraries at the start of the load library search chain then execute program MYPROG.

TOP

Syntax:

```
>>--- TOP -------------------------------------------------------------------><
```

Description:

Use the TOP command to display the top lines of the data in the focus window. The first line of the data becomes the first line of the display area.

UP

Syntax:

```
>>-- UP ----+--------------+--+-----------+---------------------------------><
            |              |  |           |
            +- windowname -+  +- CURSOR --+
                              |           |
                              +- DATA ----+
                              |           |
                              +- HALF ----+
                              |           |
                              +- MAX -----+
                              |           |
                              +- PAGE ----+
                              |           |
                              +- n_lines -+
```

Description:

Scroll the view of the data within the specified window upwards towards the top of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_lines* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, UP, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to UP is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one line is used.

By default this command is assigned to function key **PF7**.

Parameters:

windowname
The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CURSOR
The line on which the cursor is positioned becomes the last line of the scrolled display.
If the cursor is positioned outside the display area or on the first line within the display area, then UP PAGE is executed instead.

DATA
Scroll up to display one page (display window depth) less one line of data.
The first line in the current display area becomes the last line of the scrolled display.

HALF
Scroll up half a page of data.
The line that is half way down the page of data in the current display area becomes the last line of the scrolled display.

MAX
Scroll up to display the first page of data.
The first displayable line becomes the first line of the scrolled display.

PAGE
Scroll up to display the next whole page of data.
The line before the first line of the current display area becomes the last line of the scrolled display.

n_lines
Scroll up a specified number of lines.
The line that is *n_lines* lines above the current line becomes the first line of the scrolled display.

VCAT

Syntax:

```
>>-- VCat --+-------------------+--------------------------------------------><
            |                   |
            +--- cblv_syntax ---+
            |                   |
            +-- < -- cblv_ctl --+
```

Description:

Use the VCAT command to open the Execute CBLVCAT window and optionally execute CBLVCAT control statements.

The Execute CBLVCAT window may also be opened via the File menu of the SELCOPY/i main window menu bar.

Parameters:

`cblv_syntax`

Valid CBLVCAT syntax to be executed when the Execute CBLVCAT window is opened. Refer to the CBLVCAT User Manual for command reference.

Note: The separator character, which by default is "!" (exclamation mark), may be used to enter multiple CBLVCAT operations on a single control statement.

Currently, CBLVCAT control statements are restricted to a maximum length of 71 characters.

This parameter is placed in the first VCAT command line field of the Execute CBLVCAT window.

`<`

CBLVCAT input control statements will be passed from the data set referenced by cblv_ctl, to the Execute CBLVCAT window for CBLVCAT execution.

`cblv_ctl`

The data set name of an MVS sequential data set or PDS and member, VSE LIBR lib.sublib and member or CMS fileid containing CBLVCAT control statements.

This parameter, together with <, is placed in the first VCAT command line field of the Execute CBLVCAT window.

Examples:

`V q cblname`

Generate CBLVCAT Query CBLNAME report.

`V option pw 133 !report vcat dsn type component alloc3 !lc key=cbl type=c`

Generate CBLVCAT catalog report.

`V < CBL.VVC.CTL(REPVTOC)`

Generate CBLVCAT report by executing control statements located in PDS member REPVTOC of CBL.VVC.CTL.

`V < PRD2.CBLVCAT.REPVTOC.CTL`

Generate CBLVCAT report by executing control statements located in VSE LIBR member REPVTOC of sublibrary PRD2.CBLVCAT.

VIEW

Syntax:

```
>>-+- View ---+--+----------------------------------------------------------+--><
   |          |  |                                                          |
   +- Browse -+  +- fileid --+----------------------------------------------+
     (1)                     |                                              |
                             +- ( -+- PROFILE macroname -+-+----------------+
                                   |                     | |                |
                                   +- NOPROFile ---------+ +-| HFS Opts |-+
```

Notes:

1. BROWSE is a synonym of VIEW for VSE and CMS systems only.

Description:

Use the VIEW command to open a CBLe text edit window view to edit a file in read only mode.

If the CBLe text editor main window has been stopped, VIEW will start the CBLe main window before opening the read only edit view of the requested file.

A file edited in read only mode has **(Read Only)** in the title bar of the document window view.

Read-write edit may be invoked using the SELCOPY/i EDIT command.

Parameters:

`fileid`

The fileid of the file to be viewed.

For MVS, *fileid* may be the DSN of a sequential or VSAM data set, the member name (with or without the library DSN) of a PDS/PDSE member or an HFS file name. If member name is specified without a library DSN, the DSN of the library member in the current edit view is used.

For VSE, *fileid* is the member name of a LIBR library in lib.sublib.mn.mt format.

For CMS, *fileid* is a CMS fileid.

Note that if **fileid** is not specified, then EDIT is executed for the fileid specified by the SELCOPY/i INI variable System.CmdTEXT. If System.CmdTEXT has not been set, then no action is taken.

`PROFILE` *`macroname`*

The REXX edit macro to be executed as the profile when editing the file.

This macro overrides use of the default profile macro defined by the SELCOPY/i INI option Edit.DefProfile=*macroname* and/or the CBLe command SET DEFPROFILE (default PROFILE.)

The macro name must exist in a library within the CBLe macro path.

The PROFILE option only affects the profile for the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the edit ring later in your edit session.

`NOPROFILE`

Suppresses use of a profile macro when editing the file.

The NOPROFILE option only affects the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the ring later in your edit session.

HFS Opts

See CBLe CLI command EDIT for supported HFS parameters.

VOLSTATS

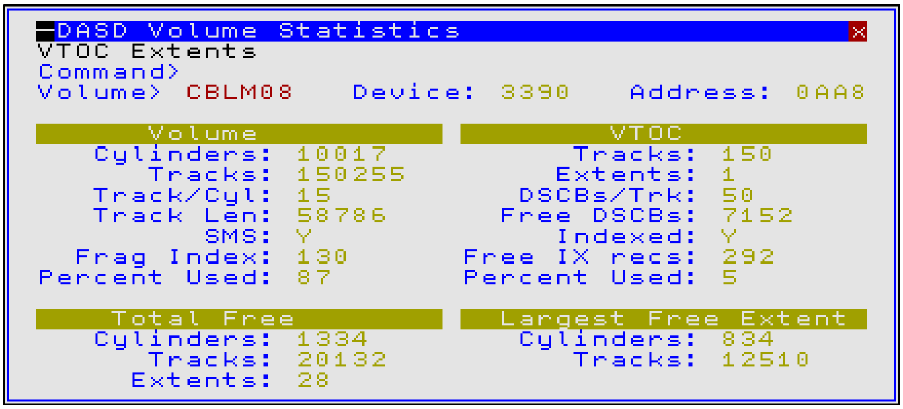
Syntax:

```
>>--+- VOLSTATS -+--+------------+------------------------------------------><
    |            |  |            |
    +- VOL ------+  +-- volser --+
```

Description:

Use the VOLSTATS command to open a DASD Volume Statistics window and optionally display statistics for a specific DASD volume.

The DASD Volume Statistics window may also be opened via the prefix command **V** on any list window entry containing a volser field.



```
 DASD Volume Statistics                                      X
VTOC Extents
Command>
Volume> CBLM08    Device: 3390    Address: 0AA8

          Volume                            VTOC
   Cylinders: 10017                   Tracks: 150
      Tracks: 150255                 Extents: 1
   Track/Cyl: 15                   DSCBs/Trk: 50
   Track Len: 58786               Free DSCBs: 7152
         SMS: Y                      Indexed: Y
  Frag Index: 130                Free IX recs: 292
Percent Used: 87                Percent Used: 5

        Total Free                 Largest Free Extent
   Cylinders: 1334                 Cylinders: 834
      Tracks: 20132                   Tracks: 12510
     Extents: 28
```

Figure 135. DASD Volume Statistics Window.

Parameters:

volser

The volume serial number of the volume for which statistics are to be displayed.
This parameter is placed in the Volume field of the DASD Volume Statistics window.

Examples:

```
 VOLSTATS  Z2RES1
```

WINDOWLIST

Syntax:

```
>>--+- WINDOWLIST -+-------------------------------------------------------><
    |              |
    +- WL ---------+
```

Description:

Use this command to open the Window List window which lists all open windows.

The Window List window may also be opened via the **Window** item of the SELCOPY/i main window menu bar.

Position the cursor on an entry in the window list and hit <Enter> or, if configured, double-click the left mouse button on the list entry, to close the window list and make the selected window the focus window.

WINDOWNAMES

Syntax:

```
>>--+- WINDOWNAMES -+------------------------------------------------------><
    |               |
    +- WN ----------+
    |               |
    +- NAMES -------+
```

Description:

This command toggles the display of window names in the title bar. When the window names are displayed, they are shown left justified in the title bar followed by a colon.

Unix System Services (USS) Commands

The CBLe text editor and SDE (Structured Data Environment) Edit support HFS files and the concept of a current working directory. This enables users to reference HFS files by an HFS path relative to the current working directory.

To fully support this functionality and assist with HFS file management for data edit, certain UNIX System Services commands are supported as part of the SELCOPY/i CLI command set. These commands are prefixed by "USS".

USS prefixed SELCOPY/i commands may only affect HFS path names and so specification of "/" (slash) within the path name or a leading "." (dot/period) in order to identify the fileid as an HFS path name is unnecessary.

Command	Description
USS CHDIR	Change the current working directory.
USS GETCWD	Display the current working directory.
USS LINK	Define a new HFS hard link to a file.
USS MKDIR	Define a new HFS directory.
USS REALPATH	Display the absolute HFS file path for a given relative HFS path.
USS RENAME	Rename an existing HFS file, hard link, symbolic link or directory.
USS RMDIR	Remove an existing, empty HFS directory.
USS STAT	Display status of a specified HFS path.
USS UNLINK	Remove an existing HFS file, hard link or symbolic link.

USS CHDIR

Syntax:

```
>>-- USS ----- CHDIR ------ hfs_path ----------------------------------------><
```

Description:

Change the current working directory.

USS CHDIR is equivalent to the USS shell command CD but without the additional options.

Parameters:

hfs_path
An HFS path name representing a directory.

USS GETCWD

Syntax:

```
>>-- USS --+-- GETCwd --+----------------------------------------------------><
           |            |
           +--- PWD ----+
```

Description:

Display the current working directory. If executed from a CBLe or SDE edit view, output is displayed on the message line. Otherwise, output is displayed in a popup message window.

USS GETCWD is equivalent to the USS shell command PWD.

Parameters:

USS GETCWD has no parameters.

USS LINK

Syntax:

```
>>-- USS ----- LINK ----- old_hfs_path ---- new_hfs_path -------------------><
```

Description:

Create a hard link to an existing HFS file.

USS LINK is equivalent to the USS shell command LINK.

Parameters:

old_hfs_path
An HFS path name representing a file. This may be the HFS file name, another hard link or a symbolic link. If *old_hfs_path* is a symbolic link, a hard link is created to the file that results from resolving the symbolic link.

new_hfs_path
The HFS path name of the new hard link to the file data.

USS MKDIR

Syntax:

```
>>-- USS ----- MKDIR ------ hfs_path ----------------------------------------><
```

Description:

Create a new HFS directory.

USS MKDIR is equivalent to the USS shell command MKDIR but without the additional options.

Parameters:

hfs_path
An HFS path name representing a directory.

USS REALPATH

Syntax:

```
>>-- USS ----- REALPATH --- hfs_path ----------------------------------------><
```

Description:

Display the absolute HFS path name for the specified (relative) HFS path name.

Parameters:

hfs_path
Any HFS path name.

USS RENAME

Syntax:

```
>>-- USS ----- RENAME --- old_hfs_path ---- new_hfs_path -------------------><
```

Description:

Rename an existing HFS file, hard link, symbolic link or directory name.

USS RENAME is equivalent to the SELCOPY/i RENAME command except that rename arguments are always treated as HFS path names.

Parameters:

old_hfs_path
An HFS path name representing a file, hard link, symbolic link or directory name.

new_hfs_path
The new HFS path name.

USS RMDIR

Syntax:

```
>>-- USS ----- RMDIR ------ hfs_path ----------------------------------------><
```

Description:

Remove an existing, empty HFS directory.

USS RMDIR is equivalent to the USS shell command RMDIR except that, currently, no option exists to remove intermediate directory components.

Parameters:

hfs_path
An HFS path name representing a directory.

USS STAT

Syntax:

```
>>-- USS ----- STAT ------- hfs_path ----------------------------------------><
```

Description:

Display the status of the specified HFS path name.

This includes the absolute HFS path name, type, file size, blocksize, format and permissions (octal).

Parameters:

hfs_path
An existing HFS path name.

USS UNLINK

Syntax:

```
>>-- USS ----- UNLINK ----- hfs_path ----------------------------------------><
```

Description:

Unlink the specified HFS path name.

USS UNLINK is equivalent to the USS shell command UNLINK.

Parameters:

hfs_path
An existing HFS path name representing a file name, hard link or symbolic link.
Alternate path names to the same data are unaffected.

SELCOPY/i VTAM commands

Commands may be passed to the SELCOPY/i VTAM application via the system operator console.

In MVS, this is achieved using the MODIFY (F) JES command and the appropriate job name as follows:

```
  MODIFY   CBLIVTAM,command
```

In VSE, this is achieved via an operator communications (OC) exit using the attention routine (AR) command MSG for the partition running SELCOPY/i VTAM. e.g.

```
  MSG      F8,DATA=command
```

Command	Description
MESSAGE	Send a text message to one or more users logged on to SELCOPY/i VTAM.
QUERY	Query the SELCOPY/i VTAM environment.
STOP	Stop SELCOPY/i VTAM.

MESSAGE

Syntax:

```
>>--+- MESsage -+--+- user -+--- text ---------------------------------------><
    |           |  |        |
    +- MSG -----+  +-- * ---+
```

Description:

Send a text message to a single user or all users logged on to SELCOPY/i VTAM. The message text will appear in a pop-up window at the user's terminal when a 3270 AID key is hit. (e.g. <Enter>, any PFKey, etc.)

Parameters:

user

The user id of the user to whom the message is to be sent.
If "*" (asterisk) is specified, then the message is sent to all users who are logged on to SELCOPY/i VTAM.

text

The message text.

Examples:

```
F CBLIVTAM,MSG JOHNB Please browse CBL.CMX(SKEL).

MSG F8,DATA=MESSAGE * Please logoff. SELCOPY/i VTAM will be stopped at 10:00.
```

QUERY

Syntax:

```
             +- Users -+
             |         |
>>-- Query --+---------+------------------------------------------------------><
```

Description:

Query information about the SELCOPY/i VTAM environment. SELCOPY/i currently supports only one parameter (i.e. USERS) which identifies all users logged on to SELCOPY/i VTAM.

Parameters:

USERS

Display information about users who are logged on to SELCOPY/i VTAM.

Examples:

```
MSG F8,DATA=Q
 ZZSV021I Applid CBLIVTAM has 2 active sessions
         User     Terminal Session
         JGE1     D20001   30000002
         NBJ1     D20101   12000003
```

STOP

Syntax:

```
>>-- STOP -------------------------------------------------------------------><
```

Description:

Stop the SELCOPY/i VTAM job.

Examples:

```
F CBLIVTAM,STOP

MSG F8,DATA=STOP
```

SELCOPY/i Dump Files

SELCOPY/i dump files are supported for SELCOPY/i running in MVS environments only.

In order to assist CBL software engineers to correct any defects encountered in the SELCOPY/i system and programs, SELCOPY/i dump files exist to store formatted storage dumps.

By default, the **System.AbendTrap** variable is set **ON** in the SELCOPY/i System INI file. Therefore, in the event of a program check or program abend occuring which ultimately halts the CBL interactive environment, a message is sent to the user and control is passed to SELCOPY/i's abend handler routines.
If AbendTrap is set **OFF**, any abnormal program end is handled by the operating system.

Each time the SELCOPY/i abend handler is called, a new dump file is allocated with DSN prefix qualifier(s) determined by the **System.DumpDSNPrefix** variable in the SELCOPY/i User or System INI file.
The remainder of the dump file DSN is of the form **.Dyyyyddd.Thhmmssx**, representing the current local date and time.

Dump files are allocated as physical sequential data sets with DCB=(RECFM=VB,LRECL=256,BLKSIZE=0) and SPACE=(CYL,(9,5)).

If an abend is encountered in SELCOPY/i, then please contact the CBL support desk via telephone on +44 1656 650692 or via email at support@cbl.com. A request to email the SELCOPY/i dump file to CBL file is likely.

Appendix A - SELCOPY/i Window Classes

The following table identifies the SELCOPY/i Windows Classes.

Window Class Name	Window Class Description
VCIWMAIN	SELCOPY/i main window
EDTWMAIN	CBLe main window
EDTWEDIT	Text Edit document window
EDTWFIND	Text Edit FIND dialog window
EDTWCHNG	Text Edit CHANGE dialog window
EDTWSORT	Text Edit SORT dialog window
EDTWFILL	Text Edit FILL dialog window
EDTWEMSG	Text Edit message window
EDTWHEXE	Text Edit line Hex Dump view
SDEWVIEW	Structured Data Edit document view
LISTFRAM	List window
LISTFILE	List File window
VCIWEXEC	CBLVCAT Interactive window
SDBWDBUG	SELCOPY Debug main window
SYSIN	SELCOPY Debug SYSIN Text Edit document view
SYSPRINT	SELCOPY Debug SYSLST/SYSPRINT Text Edit document view
WTOLOG	SELCOPY Debug WTO LOG Output Text Edit document view
SQLLOG	SELCOPY Debug DB2 SQL LOG Output Text Edit document view
TRACE	SELCOPY Debug TRACE Output Text Edit document view
STORAGE	SELCOPY Debug POS Hex Dump view
HTMWMAIN	Help window
VCIWDEFA	Allocate New NonVSAM Cataloged Dataset dialog window
VCIWDEFC	Define VSAM Cluster dialog windows
VCIWDFAL	Define VSAM Catalog ALIAS dialog window
WINWIPO0	Interactive Panel windows
WINWALID	Define PDS/PDSE Member ALIAS dialog window
WINWIEBC	Execute IEBCOPY dialog window
WLDIALOG	Window List window
SDEWFCOP	File Copy dialog window
CALCULAT	Rexx Calculator window
CALENDAR	Calendar window
HEXDUMP	Storage display window
SYSINFO	System Information window

Appendix B - List File Prefix Command Summary

See List Window Prefix Area for a description of the list window class prefix area and its features.

The following table is a summary of all the standard prefix commands supported by list data object windows, the Execute CBLVCAT window and File Search window.

A Open the Create Library Alias dialog window for Library Lists or the Define Catalog ALIAS for all other file lists.

AP Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.

AS Open an Associations list window for the entry.

B Open the CBLe text editor to to perform SDATA BROWSE on the entry.

C Open the File Copy dialog panel to copy the entry.

CF Open the Compare Files Panel for this entry, using the entry name as the New File field entry.

CL Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.

D Delete the entry. User will be prompted to verify the deletion.

E Open the CBLe text editor to edit the entry.

EU Open the SDE structured data editor to edit the entry in update mode only.

EX Execute the library member entry. (Invokes the TSO command, EXECUTE, using the entry name as input.
Supported in MVS TSO or ISPF environments only.

F Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
Supported for MVS SELCOPY licensees only on all types of data set.

FO Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
Supported for MVS SELCOPY licensees only.

FS Open the File Search window to search the contents of the entry.
Supported for MVS PDS/PDSE, CMS fileid, VSE LIBR sub-library and member entries only.

I Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.

IC Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.

J Submit the library member entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
Supported in MVS and VSE environments only.

K Delete (Kill) the entry without prompting for verification.

L Open a Dataset List window for the entry.
Supported for Execute CBLVCAT windows only.
For **VSE LIBR Library member** list windows only, lock the LIBR member.

M Open a Library List window for the entry.
Supported for MVS PDS/PDSE, VSE LIBR library and sub-library entries only.

Q List dataset enqueues (major name SYSDSN) for the entry.
Supported for MVS only.

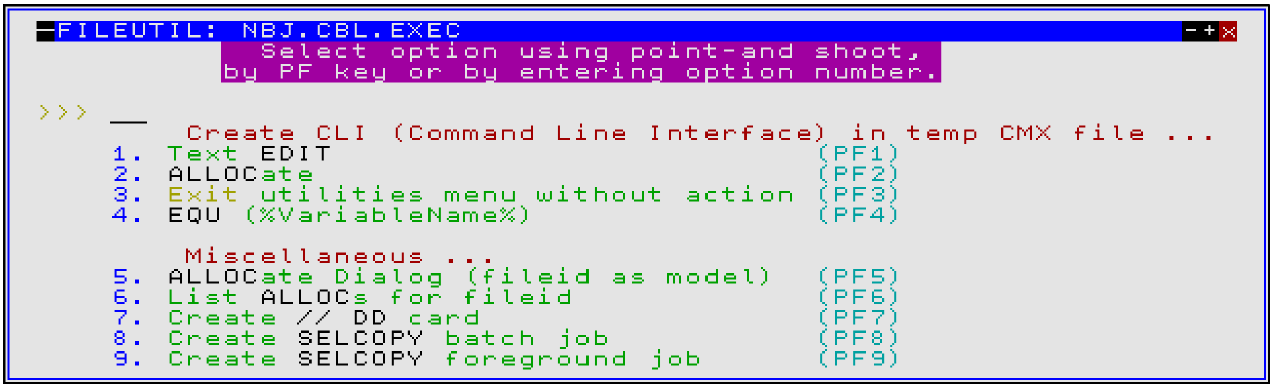
R Rename the entry.

SD Open the SDE BROWSE/EDIT Dialog Window to edit or browse the entry's data within a Structured Data Environment window view.
Supported for MVS SELCOPY licensees only.

T Issue a LISTVCAT operation against the entry with parameters TUNE and DEFINE.
For DASD Listwindows only, open the VTOC list window for the volume entry.

U Unallocate the MVS DD name or UNLOCK the VSE LIBR member entry.
Entries may only be unallocated or unlocked by the user that originally allocated or locked it.

UT Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.



```
 FILEUTIL: NBJ.CBL.EXEC                                                  -+x
                     Select option using point-and shoot,
                   by PF key or by entering option number.

 >>> ___
         Create CLI (Command Line Interface) in temp CMX file ...
     1.  Text EDIT                                 (PF1)
     2.  ALLOCate                                  (PF2)
     3.  Exit utilities menu without action        (PF3)
     4.  EQU (%VariableName%)                      (PF4)

         Miscellaneous ...
     5.  ALLOCate Dialog (fileid as model)         (PF5)
     6.  List ALLOCs for fileid                    (PF6)
     7.  Create // DD card                         (PF7)
     8.  Create SELCOPY batch job                  (PF8)
     9.  Create SELCOPY foreground job             (PF9)
```

Figure 136. File Utilities Menu.

Options are selected by entering the required option number at the command prompt or executing the equivalent PFKey.

On selecting one othese options, a dialog panel or edit view containing generated syntax for the selected entry *entry_name* is opened as follows:

1. Text Edit	`<edit 'entry_name'`
2. ALLOCate	`<alloc f(MYDDNAME) reuse shr dsn('entry_name')`
4. EQU (%VariableName%)	`<equ MyFile 'entry_name'`
5. ALLOCate Dialog (fileid as model)	Opens the Allocate NonVSAM dialog.
6. List ALLOCs for fileid	`LA; where DsN=entry_name`
7. Create // DD card	`//MYDDNAME DD DISP=SHR,DSN=entry_name`

8. Create SELCOPY batch job	//FILEUTIL EXEC PGM=SELCOPY
	//MYDDNAME DD DISP=SHR,DSN=*entry_name*
	//SYSPRINT DD SYSOUT=*
	//SYSIN DD *
	option worklen=65536 NoRdw
	read MYDDNAME
	print len=100 type=b stopaft=22
	/*

9. Create SELCOPY foreground job	** %USER%.FILEUTIL.Tnnnnnn.SLC *** L=001 --- yyyy/mm/dd HH:MM:SS (%USER%) *<RunSelc | Use this command to run SELCOPY in the foreground. option NoRdw * worklen=65536 read MYDDNAME dsn='*entry_name*' dirdata print type=b stopaft=22

V Open the CBLe text editor to View (edit read/only) the entry.

VC Open an Execute CBLVCAT window and issue a LISTVCAT and/or LISTVTOC operation (as appropriate) for the entry.

Z Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
Supported in MVS environments only.

? Open the DASD Volume Statistics window for the volume in the list entry.

/ Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command.
Assigned to PF4 by default.

> Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns.
Assigned to PF2 by default.

Command Cross-Reference

	Prefix Commands																														
	A	**AS**	**AP**	**B**	**C**	**CF**	**CL**	**D**	**E**	**EU**	**EX**	**F**	**FO**	**FS**	**I**	**IC**	**J**	**K**	**L**	**M**	**Q**	**R**	**SD**	**T**	**U**	**UT**	**V**	**VC**	**Z**	**?**	**/**	**>**
VCAT	Y	Y	-	Y	Y	-	-	Y	Y	-	-	Y	Y	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	-	Y	Y	Y	Y	-	Y
LVR	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	Y	Y	Y	Y	-	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	-	Y	Y
LVOL	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	Y	-	Y	Y	Y
LC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
LD	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
LV	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y
LVX	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y
LA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y
Lab	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	Y	Y
LL	**2**	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	**1**	**1**	-	Y	Y	-	**1**	Y	Y	-	-	-	Y	Y
LQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y
LJQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y
LAS	Y	Y	-	Y	Y	Y	-	Y	Y	Y	-	Y	Y	-	Y	-	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	-	-	Y	Y
LP	-	-	-	Y	Y	Y	-	Y	Y	Y	-	Y	-	-	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	Y	-	-	-	Y	Y
FS	**2**	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	**1**	**1**	-	Y	Y	-	**1**	Y	Y	-	-	-	Y	Y

Legend:

1. VSE LIBR member list only.
2. MVS LIBR member list only.

Glossary

The following is a glossary of terms used in this document.

3270 Emulator
Third party software that emulates Mainframe 3270 hardware terminals on PC and UNIX based platforms.

CLI
A Command Line Interface is a text based method by which users can execute functions supported by the application.

CBLe
A powerful text editor that runs as an MDI application under SELCOPY/i. CBLe supports its own command line interface (CLI) and has been developed based on specifications for IBM's ISPF Edit, CMS XEDIT and Mansfield Software's KEDIT for Windows.

CBLVCAT
CBL licensable product that supports VSAM file tuning and VTOC, ICF/VSAM catalog and VSE LABEL reporting. Executes as a batch facility or interactively as a SELCOPY/i application.

Edit View or **Text Edit View**
A CBLe MDI document window that contains a display of text edited data. If the same file is displayed in multiple windows, then the user has multiple edit views of the file. Each edit view can have a different current line, ARBCHAR setting, ZONE columns, etc.

FDB
A Field Descriptor Block used to define the field column elements of a list.

List Column
A single column of text within the display area of the current list window. A list column may fall within a List Field Column or in the gap between field columns.

List Current Column
The first scrollable list column within the display area of the current list window. Key list columns (FDB field Key=Yes) are non-scrollable and so are not included

List Current Row
The first visible list row within the display area of the current list window.

List Field Column
A single column field within the current list window which has a maximum length as defined by the field's FDB entry.

List Focus Column
The list column on which the cursor is positioned within the list focus row.
If the cursor is positioned outside the list display area (e.g. the command line) or within the list prefix area, the list focus column is defined as being the list current column.

List Focus Row
The row within the current list window on which the cursor is positioned. If the cursor is positioned outside the list display area (e.g. the command line), then the list focus row is defined as being the list current row.

List Window
A class of SELCOPY/i window containing rows of associated information. List windows support point-and-shoot column sorting; select, sort and filter CLI commands; and prefix area commands.

MDI
Multiple Document Interface is a Microsoft specification for PC applications that enable the user to work with multiple documents at the same time. Each document is displayed in a separate child window within the client area of the application's main (frame) window. Typical MDI applications on PCs include word-processing and spread sheet applications.

MDI Client Area window
The MDI client area window is the display area within an MDI application's frame window. The MDI client area serves as the backround for MDI child windows.

MDI Child/Document Window
An MDI child or document window is opened in an application's client area window each time a document is opened. Each child window has a sizing border, title bar, window menu, minimise, maximise, restore and close buttons. A child window is clipped so that it is confined to the client window and cannot appear outside it.
When a child window is maximized, its client area completely fills the MDI client area window. In addition, the system automatically hides the child window's title bar, and adds the child window's window menu icon and Restore button to the MDI application's menu bar.

MDI Frame Window
An MDI frame window may be considered the main window of an MDI application. It is the parent window of the MDI client area window in which MDI child windows are opened. It has a sizing border, title bar, window menu, minimise, maximise restore and close buttons.

Modal Window
A modal window requires user interaction before further processing can occur. Window focus cannot be placed on any other window until the modal window is closed.

Edit Ring
The set of all **files** being edited within CBLe. It is not the set of all windows opened. e.g. The contents of one file may be displayed in more than one edit view (window.)

SDB
See SELCOPY Debug.

SELCOPY Debug (SDB)
An Intergrated Development Environment for SELCOPY that runs as an MDI application in SELCOPY/i.

SELCOPY/i
The interactive environment developed by CBL and supplied as part of the SELCOPY Product Suite. Requires licence key for SELCOPY and/or CBLVCAT elements of SELCOPY Product Suite.

SELCOPY/i INI
File containing configuration options for SELCOPY/i. The SELCOPY/i System INI file is processed on startup of SELCOPY/i and contains options that apply to all users. The SELCOPY/i User INI file contains options specific to each user that may, where appropriate, override options set in the SELCOPY/i System INI file.

SELCOPY/i VTAM
Name of the multi-user version of the SELCOPY/i application that executes under VTAM.

Storage Display Window
A SELCOPY/i window containing hexadecimal and character display of areas of storage.